CONNECTIVITY CONSTRAINTS IN NETWORK ANALYSIS

A Dissertation

by

YIMING WANG

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

DOCTOR OF PHILOSOPHY

Chair of Committee,	Sergiy Butenko
Committee Members,	Erick-Moreno Centeno
	Kiavash Kianfar
	Anxiao Jiang
Head of Department,	César O. Malavé

August 2015

Major Subject: Industrial & Systems Engineering

ABSTRACT

This dissertation establishes mathematical foundations of connectivity requirements arising in both abstract and geometric network analysis. Connectivity constraints are ubiquitous in network design and network analysis. Aside from the obvious applications in communication and transportation networks, they have also appeared in forest planning, political distracting, activity detection in video sequences and protein-protein interaction networks. Theoretically, connectivity constraints can be analyzed via polyhedral methods, in which we investigate the structure of (vertex)-connected subgraph polytope (**CSP**).

One focus of this dissertation is on performing an extensive study of facets of **CSP**. We present the first systematic study of non-trivial facets of **CSP**. One advantage to study facets is that a facet-defining inequality is always among the tightest valid inequalities, so applying facet-defining inequalities when imposing connectivity constraints can guarantee good performance of the algorithm. We adopt lifting techniques to provide a framework to generate a wide class of facet-defining inequalities of **CSP**. We also derive the necessary and sufficient conditions when a vertex separator inequality, which plays a critical role in connectivity constraints, induces a facet of **CSP**. Another advantage to study facets is that **CSP** is uniquely determined by its facets, so full understanding of **CSP**'s facets indicates full understanding of **CSP** itself. We are able to derive a full description of **CSP** for a wide class of graphs, including forest and several types of dense graphs, such as graphs with small independence number, s-plex with small s and s-defective cliques with small s. Furthermore, we investigate the relationship between lifting techniques, maximum weight connected subgraph problem and node-weight Steiner tree problem and study the computational complexity of generation of facet-defining inequalities.

Another focus of this dissertation is to study connectivity in geometric network analysis. In geometric applications like wireless networks and communication networks, the concept of connectivity can be defined in various ways. In one case, connectivity is imposed by distance, which can be modeled by unit disk graphs (**UDG**). We create a polytime algorithm to identify large $2-$*clique* in **UDG**; in another case when connectivity is based on visibility, we provide a generalization of the *two-guard* problem.

ACKNOWLEDGEMENTS

I have been truly blessed to have Dr. Sergiy Butenko as my advisor. I would like to express my deepest gratitude to Sergiy for his insightful ideas and excellent guidance in my professional development, and his patience that provides me with an excellent atmosphere for doing research. I am grateful to Sergiy for making my doctoral experience a memorable one.

I would like to thank my committee, including Dr. Moreno, Dr. Kianfar and Dr. Jiang, for helpful feedback and support during my studies. I would also like to thank Dr. Yates for his help in collecting data for network analysis.

I would like to thank Austin Buchanan and Jeff Pattillo, my collaborators, for their valuable ideas during discussions.

I would like to thank the ISE department for financially supporting my graduate studies and providing me with a wonderful learning atmosphere and opportunities to develop my skills in both academia and industry. I would especially like to thank Dr. Guy Curry, Andrew Johnson and Yu Ding for providing me with opportunities to teach, and their guidance and support. I would also like to thank the friendly administrative and technical staff at ISE, especially Judy Meeks, Erin Roady, Michele Bork, Mark Henry, Mark Hopcus and Dennis Allen.

I would like to thank Bo Wei, Ang Li, Yi Zhang, Su Zhao, Chitra Balasubramanian, Samyukta Sethuraman, Zeynep Ertem, Oleksandra Yezerska, Je Sang Sung, Anurag Verma, Shahram Shahinpour, Yu Fu, Xin Ma, Shiyan Yang, and countless others for the friendship we shared during my time in College Station.

More than anyone I would like to thank my parents for the love they gave me throughout my life. This accomplishment belongs as much to them as it does to me. Words cannot express the love and pride I have for my mother, especially for her solid support and guidance of my life after my father passed away during my graduate study, and words cannot express my deepest memory of my father either. There is no way to repay that kind of love and support.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE Page

1. INTRODUCTION

Network is a popular mathematical object comprised of a set of nodes with edges representing interaction between nodes. Every group of objects that have possible mutual relationship can form a network. For example, a group of people is a network as they may know each other; a set of twitter accounts is a network as they may follow each other or retweet others' comments; a group of computer is a network as they may connect and communicate to each other in order to finish one task; the vehicles in the same road can be recognized as a network as well because they follow the same traffic and may have accidental interaction like crashes. In summary, networks arise everywhere in our society. A mathematical term that is closely related to and is often used to represent the concept of a network is a *graph*. A graph is defined to be a pair of sets $G = (V, E)$, where V is a set of vertices, which represents nodes in a network, and E is the set of edges between nodes, which represents the pairwise interactions between nodes. Representing information as a graph allows for interrelated data to be gathered concisely and in a global context, so analysis of a graph allows us to search the global properties that cannot be easily observed using only local information.

Connectivity is such a global property. A graph G is said to be connected if for every pair of its vertices there exists a path connecting them. Given a graph G with at least two vertices, the connectivity of G is defined as the minimum number k of vertices that need to be removed in order for the remaining graph to become disconnected or have just one vertex remaining. Connectivity plays a critical role in network design and network analysis problems, and its role varies in different applications.

In some applications, it is necessary to ensure at least 1-connectivity (connectivity for short in the case that no ambiguity occurs), and enhancing the connectivity of the network may be strongly beneficial. Two examples are the design of telecommunication networks and transportation networks. It is essential to guarantee connectivity in telecommunication network design applications like design of a computer network or a telephone network, because the spread of information throughout the whole network needs to be guaranteed. A network with high connectivity can survive and function normally when some nodes fail, so in real-life applications such a network is reliable and thus it is also important to ensure high connectivity. A great amount of research was carried out towards designing a reliable telecommunication network with high connectivity, which is summarized in [37, 34, 59, 42]. In construction of a transportation network, connectivity is also a major concern. One reason is that it is an important task to ensure accessibility of every node. so it is necessary to ensure connectivity; another reason is that we need to distinguish between nodes with high connectivity and those with low connectivity and put high capacities in nodes with high connectivity to keep the network robust. Research related to connectivity in transportation networks includes [47, 50, 70, 85].

In some applications, the focus is only to ensure 1-connectivity. In [21], a forest harvest scheduling problem was studied. One objective of the harvest scheduling was to maintain large contiguous patches of mature forest in order to protect wildlife habitats. The authors considered the network consisting of small patches of forest as nodes and pairs of adjacent patches as edges, so it was sufficient to ensure connectivity of this network in order to achieve their objective of wildlife protection. Another application of connectivity requirements is in political districting [36]. Political districting is the process by which an area (e.g., a state) is partitioned into smaller districts with almost the same population of eligible voters. This problem is seemingly unrelated to connectivity, but connectivity requirements are in fact essential in solving this problem, because the realistic methodology is to split the area into very small pieces ignoring population constraints first and then gather small pieces together to form districts with roughly the same population. Because Federal laws require the districts to be contiguous, it is a major task to ensure connectivity in the step of generating districts.

In other applications, connectivity of the whole network is not required, but it is important to identify the parts of network that are connected. A mathematical term to represent the connected part is *connected component*, which is defined as a maximal by inclusion connected subgraph of the original graph. Identifying connected components has important applications in the field of computer vision. For example, one task in image recognition is to detect connected components in binary digital images [86]. Additionally, when the detection of connected components is integrated into a more complicated system like human-computer interaction interface system, research shows more hidden information is derived [24, 95]. Furthermore, in [23], an application of connectivity in recognition of unusual activities in video sequences was introduced. Recognition of connected subgraphs (not necessarily the connected components) is also useful in biology. In [30], a protein-protein interaction network was analyzed. With some scoring on nodes, the authors developed an algorithm to find the connected subgraphs with large node scores and asserted they are functional modules with high probability.

Aside from ubiquity in practice, connectivity requirements are also closely related to many combinatorial optimization problems in theory. The well-known Steiner tree problem asks for a shortest connected network which spans a given set of points. Details on this problem can be found in [52]. Our research shows that there are close connections between connectivity requirements and the Steiner tree problem. Other examples include the maximum sub-array problem [10] and maximal sums problem [18]. In computer science, the maximum subarray problem asks for the contiguous subarray within a one-dimensional array of numbers which has the largest sum, while the maximal sums problem is to find the sub-vector with the largest sum in a sequence of numbers. Both of them can be transformed to network optimization problems, where we search for the connected subgraphs with additional desired properties. In addition, connectivity is directly related to graph clustering problems like the problems of finding cliques and clique relaxation structures. A clique is a set of vertices in a graph for which every pair of vertices is directly connected by an edge. It has the highest connectivity among graphs with the same number of vertices. In [81] it is shown that one way to relax the concept of a clique is to relax the connectivity requirements.

1.1 Study of Connectivity

There are two major methods to analyze connectivity requirements. One method is based on the concept of *flow network*. A flow network is a directed graph where each edge or vertex has a capacity and receives a flow. The flow must satisfy the restriction that the amount of flow into a vertex equals the amount of flow out of it, unless it is a source vertex, which can have more outgoing flow, or sink vertex, which can have more incoming flow, and the amount of flow on an edge or a vertex cannot exceed its capacity. Net flow in the flow network is defined to be the amount of flow coming from the source vertex. Replacing each edge of the graph by two directed edges and setting adequate capacities of vertices and edges, the graph is connected if and only if the corresponding flow network accepts positive net flow when choosing source and sink vertices arbitrarily. Therefore, connectivity requirements can be represented by a set of positivity requirements of net flows under flow restrictions. Consult [34] for an excellent survey that provides a comprehensive overview of connectivity analysis via flow network method.

The other method is based on the concept of *cuts*. A vertex cut of a connected graph G is a set of vertices whose removal renders G disconnected. Analogously an edge cut of G is a set of edges whose removal renders G disconnected. In cut-based method, the connectivity requirements is imposed by making sure that a cut is not completely excluded from the set. The process is operated by adding cut-based inequalities to the model and it can be conveniently integrated into branch-and-cut algorithms. Research towards generating good cut-based inequalities and application of these inequalities in network analysis includes [43, 41, 16, 44, 3, 13]. For both methods, connectivity requirements are imposed by connectivity constraints. Connectivity constraints are inequalities that are satisfied by all connected subgraphs but violated by some subgraphs that are not connected. In the flow network-based method, the connectivity constraints ensure that net flows of all source vertices are positive under flow restrictions, while in the cut-based method the connectivity constraints are cut inequalities. To develop effective algorithms, a connectivity constraint is expected to be tight, i.e., it could avoid the occurrence of a large amount of disconnected subgraphs. The concept of tightness is closely related to the *connected subgraph polytope*, a polytope defined by all connected subgraphs, because every facet-defining inequality of this polytope is the tightest in a sense that no other inequality is strictly stronger than it. However, there is very little understanding of this polytope currently, while most research was only focused on adding tight connectivity constraints rather than finding the tightest ones.

1.2 Geometric Networks

Geometric networks are networks with extra geometric structure. In geometric networks the distribution of edges between nodes is based on certain geometric restrictions. In other words, two nodes are adjacent if some geometric constraints are satisfied. Tools can be developed to take advantage of the extra geometric structure not necessarily exhibited by general graphs.

In one type of geometric networks, commonly used to model wireless networks, we assume that the ability of two wireless nodes in the network to communicate with each other directly completely depends on the distance between them. Such a network can be modeled by *unit disk graphs*, where vertices are given by disks of a unit diameter, and two disks are connected by an edge if they have a nonempty overlap [26].

Due to the extra properties of unit disk graphs arising from their geometric nature, some problems become much easier in these graphs than in general. In particular, the maximum clique problem, which is notoriously hard in general graphs [35], can be solved in polytime when restricted to unit disk graphs [26]. Therefore, it is interesting to find out if the geometric structure also helps in solving some clique relaxation problems in unit disk graphs.

In another type of geometric networks, connectivity of nodes can be measured by the concept of *visibility*. Let the nodes in a geometric network be given by a finite set of moving objects (guards) on a plane that also contains some obstacles. Two points are said to be visible if the line segment that joins them does not intersect any obstacles. The concept of visibility has important applications in road network surveillance, robotics, motion planning and security [38]. The networks with visibility conditions may not be easily representable by graph models as the shape of networks' boundaries and obstacles have influence on visibility. Instead, the networks are usually represented by polygons, which are closed regions made up of finite chains of straight line segments in visibility problems. *The two-guard problem* is an instance of visibility problems which asks whether two objects can move on the boundary of a polygon while being visible to each other. This problem was first introduced in [53], followed by research on multiple generalization of the two-guard problem [66, 76, 77, 89, 12, 99]. All previous work uses a polygon to represent a network. However, in some real-life applications it is not adequate to use polygons because the boundaries are usually curves. In order to simulate these situations better it is necessary to propose new concepts that approximate curved boundaries in visibility problems.

1.3 Contributions and Outline

This dissertation research deals with connectivity requirements and connectivity related problems arising in network analysis. One focus of this dissertation is on performing an extensive study of facets of the connected subgraph polytope. The linear inequalities that define facets are the tightest by polyhedral theory, thus study of those facets can be utilized with classical branch-and-cut techniques in mathematical programming in developing effective integer programming algorithms to solve network analysis problems. We further search for subclasses of networks for which we have full descriptions of their connected subgraph polytopes. In such subclasses binary constraints of variables are not necessary, so linear programming techniques can be utilized, which simplify the study of connectivity requirements significantly.

The other focus of this dissertation is to explore connectivity related problems in different geometric network models. The unit disk graph model and visibility problems are studied. Tools that take advantage of various geometric properties are developed.

Broadly, this dissertation makes the following contributions. Firstly, we have developed a framework to generate a large class of facets of connected subgraph polytope for general graphs. We have presented the necessary and sufficient conditions for one class of well-known connectivity constraints to define facets. This is done using a framework that allows to generate these constraints. Then we have analyzed the computational complexity of generation process and the relationship between the maximum weight connected subgraph problem and the node-weighted Steiner tree problem. We derived several new results for these two problems. In addition, we have provided full description of connected subgraph polytope for graph that is a forest or satisfies some edge density restrictions. Furthermore, we discuss an approximation algorithm for a clique relaxation problem called the maximum *2-clique* problem in a unit disk graph. Finally, we have generalized the two-guard problem to the case with curved boundaries.

The remainder of this dissertation is organized as follows. Chapter 2 presents key background information for our research from graph theory, complexity theory, polyhedral theory, and point-set topology. In Chapter 3, we study the polyhedral structure of connected subgraph polytope and develop a framework to generate facets utilizing the lifting technique from polyhedral theory. The relationship between the maximum weight connected subgraph problem, the node-weighted Steiner tree problem, and the process of generating facets is also analyzed. In Chapter 4, we propose full description of connected subgraph polytope for subclasses of graphs. Chapter 5 is focused on the 2-clique problem in a unit disk graph. We discuss an algorithm with a guaranteed $\frac{1}{2}$-approximation ratio for solving this problem. In Chapter 6, we generalize the two-guard problem. We introduce a new concept of curvilinear polygon to generalize the usual polygon and develop new tools to deal with the curvilinear polygons. Finally, in Chapter 7, we conclude our study and present potential directions for future research.

The research in Chapter 3 and 4 is a joint work with Austin Buchanan and Sergiy Butenko. Some results from Chapter 3 and 4 appear in working papers [94, 20]. The research in Chapter 5 is a joint work with Jeff Pattillo and Sergiy Butenko based on publication [80], which is essentially a refined and enhanced version of work that originally appeared in [78].

2. BACKGROUND

This section gives the background information necessary for this dissertation. Definitions and background needed from graph theory are presented in Section 2.1. The basic understanding from complexity theory is the focus of Section 2.2. Knowledge necessary from polyhedral theory is summarized in Section 2.3. The necessary information above convex sets and functions is given in Section 2.4.

2.1 Graph Theory

For a basic introduction to general graph theory, see [29]. For an introduction of unit disk graphs, see [26]. For an introduction of flow networks, see [1]. We only provide the notations and basic definitions necessary in this dissertation.

Throughout this work we consider a finite and simple graph $G = (V, E)$, where $V = \{1, \ldots, n\}$ and $(i, j) \in E$ when vertices i and j are adjacent, with $|E| = m$. The *order* of G is the number of vertices n, and the *size* of G is the number of edges m. We use $V(G)$ and $E(G)$ to denote the vertex set and edge set, respectively, of a graph G. When G is undirected, (i, j) and (i, j) represent the same edge, while in a directed graph, they represent different edges. Except for flow networks, we assume the graph G is undirected. A weight function W can be associated with vertices and edges of G, making G a weighted graph. Given a positive integer n and $p \in [0, 1]$, a *uniform random graph* $G(n, p)$ is a graph with n vertices where the probability that an edge exists between any two vertices is p.

A graph $G' = (V', E')$ is a *subgraph* of $G = (V, E)$ if $V' \subseteq V$ and $E' \subseteq E$. Given a subset $S \subseteq V$, *induced subgraph* $G[S]$ is derived by deleting all vertices and incident edges in $V \setminus S$ from G. The graph obtained by the deletion of a vertex i or a set of vertices S form G is denoted by $G - i$ and $G - S$, respectively. Given graphs G_1 and G_2, the corresponding union graph $G_1 \cup G_2$ is $G = (V(G_1) \cup V(G_2), E(G_1) \cup E(G_2))$. $G = (\emptyset, \emptyset)$ is called a *null graph*, and $G = (V, \emptyset)$ is called a trivial graph. G is called *complete* if all its vertices are pairwise adjacent, i.e. $\forall i, j \in V$, we have $(i, j) \in E$. The complete graph on n vertices is denoted by K_n. The *complement graph* of $G = (V, E)$ is defined by $\bar{G} = (V, \bar{E})$, where $K_{|V|} = (V, E \cup \bar{E})$. A *bipartite graph* consists of two independent sets P and Q such that all edges cross between vertices in P and Q. A *complete bipartite graph* where the size of P and Q arep, q respectively is denoted by $K_{p,q}$. In particular, the graph $K_{1,n}$ is called a *star*. A *path* on n vertices, denoted by P_n, is an ordered tuple of vertices $(p_1, \ldots, p_n)$ such that consecutive vertices are adajacent in G. A graph G where there is a path between every pair of vertices is called *connected*.

For $i \in V$, the set $N_G(i)$ of vertices adjacent to i in G is called the *neighborhood* of i, and the set $N_G[i] = N_G(i) \cup \{i\}$ is the *closed neighborhood* of i. We will simplify $N_G(i)$ and $N_G[i]$ to $N(i)$ and $N[i]$, respectively, when it is obvious what graph G we have in mind. Let $deg_G(i) = |N_G(i)|$ be the degree of i in G. The maximum and minimum vertex degree in G is denoted by $\Delta(G)$ and $\delta(G)$, respectively. Given $i, j \in V$, $d_G(i,j)$ denotes the shortest length of a path between i and j in G. In order to distinguish it from Euclidean distance in the setting of unit disk graphs, we will sometimes refer to it as the *geodesic distance* between vertices. By convention, the distance between two vertices that are not connected is infinity. The *diameter* of G is $diam(G) = \max_{i,j \in V} d_G(i,j)$. For $G = (V, E)$ and a positive integer k, the k^{th} power of G is $G^k = (V, E^k)$, where $E^k = \{(i,j)|i, j \in V, d_G(i,j) \leq k\}$.

G is called a *k-degenerate graph* if every subgraph of G has a vertex of degree at most k. The *degeneracy* (k-core number, width) of G is the minimum k such that G is k-degenerate. The *connectivity* $\kappa(G)$ of G is the minimum number of vertices whose removal results in a disconnected or trivial graph. A *clique* is a subset of vertices that induces a complete graph in G. Given a graph G, the *maximum clique problem* is to find a clique of maximum cardinality in G. The *clique number* $\omega(G)$ is the cardinality of a maximum clique in G. An *independent set*(also called *stable set* or *vertex packing*) is a subset of vertices that induce a subgraph with no edges. The maximum independent set problem to find an independent set of largest cardinality $\alpha(G)$ in G, where $\alpha(G)$ is called the *independent number* of G. Obviously, I is an independent set in G if and only if I is a clique in $\bar{G}$, so $\alpha(G) = \omega(\bar{G})$. A clique (independent set) is called *maximal* if it is not a proper subset of a larger clique (independent set).

A *dominating set* in G is a subset of vertices such that every vertex in G is either in this set or has a neighbor in this set. A dominating set is *minimal* if it does not contain a smaller dominating set and *minimum* if there is no smaller dominating set in G. The minimum cardinality of a dominating set is called the *domination number* and denoted by $\gamma(G)$. If $\gamma(G) \leq k$, G is called *k-dominated*. A *vertex cover* in G is a subset of vertices such that every edge in G has at least one end point in this set. The minimum cardinality of a vertex cover is called the *vertex cover number* and denoted by $\tau(G)$.

Unit disk graphs represent a subclass of graphs that can be realized as a set of equal radius disks in the Euclidean plane, where edges are completely determined by the distance between the centers of the disks. In the *intersection model*, two disks are connected by an edge if and only if the two disks of equal radius intersect. In the *containment* model, two disks are connected by an edge if and only if each disk covers the center of the other. Not every graph can be represented as a unit disk graph. For instance, the complete bipartite graph $K_{1,7}$ is not a unit disk graph. The intersection and containment models of unit disk graphs are equivalent, meaning they specify the same subset of graphs form the collection of all graphs. However, we will work exclusively with the containment model in this dissertation when we will analyze the 2-clique problem on unit disk graphs. All disks in a clique of a unit disk graph pairwise intersect under the containment model. This fact is crucial to our results.

Let $G = (V, E)$ be a directed graph where every edge $(u, v) \in E$ has a non-negative capacity $c(u, v)$. We assume $c(u, v) = 0$ if $(u, v) \notin E$. We distinguish two vertices: a source s and a sink t. A *flow network* is a real function $f : V \times V \to \mathbb{R}$ with the following three properties for all vertices u and v:

Capacity constraints: $f(u, v) \leq c(u, v)$.

Skew symmetry: $f(u, v) = -f(v, u)$.

Flow conservation: $\sum_{w \in V} f(u, w) = 0$ unless $u = s$ or $u = t$.

The value of flow is defined by $|f| = \sum_{v:(s,v) \in E} f(s, v)$. The *maximum flow problem* is to maximize $|f|$, that is, to route as much flow as possible from s to t.

An *s, t-cut* $C = (S, T)$ is a partition of V such that $s \in S$ and $t \in T$. The *capcity* of an s, t-cut is defined by $c(S, T) = \sum_{(u,v) \in S \times T} c(u, v)$. The *minimum s, t-cut problem* is to minimize $c(S, T)$.

Theorem 1 (Max-flow min-cut theorem). *In a flow network, the maximum amount of flow $|f|$ from s to t is equal to the minimum of s, t-cut capacity $c(S, T)$.*

2.2 Complexity Theory

In this section, we give a brief review of basic concepts from the complexity theory; see [35] for more detail.

In the theory of $\mathcal{NP}$-completeness we deal with the tractability of *decision problems*, which answer with either "yes" or "no" to questions about a given object. The framework of $\mathcal{NP}$-

completeness aims to separate decision problems with solutions that are easy to verify but difficult to compute from those that are easy to compute.

A decision problem $\mathcal{Q}$ is said to be in *class* $\mathcal{P}$ if an algorithm $\mathcal{A}$ exists that can answer it correctly in a running time polynomially bounded by its input size. Since polynomial-time algorithms are regarded to be efficient, problems in class $\mathcal{P}$ and the corresponding optimization problems are considered to be easy. However, many optimization problems are not known to belong to this class, but belong to a wider class, $\mathcal{NP}$. A decision problem $\mathcal{Q}$ belongs to the class $\mathcal{NP}$ if there exists a polynomial-time algorithm $\mathcal{A}$ that, given a solution to a yes-instance, uses this solution as a certificate to verify that this is indeed a yes-instance of the problem. Note that $\mathcal{A}$ does not need to know how to construct a solution for a given yes-instance x but only need to test its correctness, so such a $\mathcal{A}$ is called a *non-deterministic polynomial* algorithm.

It is clear that $\mathcal{P} \subseteq \mathcal{NP}$ since any decision problem in $\mathcal{P}$ is also included in class $\mathcal{NP}$ by treating the algorithm $\mathcal{A}$ as the non-deterministic algorithm required for a problem in $\mathcal{NP}$. The algorithm for problems in $\mathcal{P}$ does not need the information y to solve a yes-instance of the problem in polynomial time but can construct it.

The notion of $\mathcal{NP}$-completeness was created to identify the most difficult problems in the class $\mathcal{NP}$. Two problems can be compared for difficulty by a notion of *polynomial time reducibility.* Given two decision problems Q_1 and Q_2, Q_1 is polynomial time reducible to Q_2 if there exists a polynomial time algorithm $\mathcal{A}$ that, given an instance x to Q_1, construct an instance $\mathcal{A}(x)$ to Q_2, such that x is a yes-instance of Q_1 if and only if $\mathcal{A}(x)$ is a yes-instance of Q_2. According to this definition, if Q_2 can be solved in polynomial time, so can Q_1. Consequently, Q_1 is not harder than Q_2. This indicates that if Q_2 is in class $\mathcal{P}$, so is Q_1, and if Q_1 is intractable, so is Q_2.

A problem Π is called $\mathcal{NP}$-hard if every problem in class $\mathcal{NP}$ is polynomial time reducible to Π. If an $\mathcal{NP}$-hard problem Π belongs to class $\mathcal{NP}$, then Π is called $\mathcal{NP}$-complete. By transitivity of polynomial time reducibility, to show that a given problem Π is $\mathcal{NP}$-hard it suffices to find a known $\mathcal{NP}$-hard problem that is polynomial time reducible to Π. A compendium of known $\mathcal{NP}$-complete problems can be found in [35].

2.3 Polyhedral Theory

A combinatorial optimization problem is to find an optimal object from a finite set of objects. Combinatorial optimization problems arise frequently in science and engineering. In many such problems, although the set of objects is finite, exhaustive search is not feasible. A majority of combinatorial optimization problems are actually $\mathcal{NP}$-hard, which indicates that they may be not be able to solve efficiently. A wide array of techniques including exact algorithms, heuristic methods, approximation algorithms, randomized algorithms and global optimization techniques exist to approach different combinatorial optimization problems. Many books have been written on combinatorial optimization problems that focused on different aspects of the study.

The purpose of this section is to provide a brief review of integer programming and polyhedral techniques for solving combinatorial optimization problems. One can consult [72, 87] for more detail.

Polyhedral theory allows one to use linear programming (LP) methods for solving integer programming and combinatorial optimization problems. The LP problem can be stated as follows:

$$\max\{c^T x | Ax \leq b\},$$

where the column vector $c \in \mathbb{R}^n$, $b \in \mathbb{R}^m$ and matrix $A \in \mathbb{R}^{m \times n}$ are given. The feasible region defined by the solution sets of a LP problem is called a *polyhedron*. P is said to be a *rational polyhedron* if A and b are rational; it is called *integral* if its extreme points are integral vectors. A bounded polyhedron is called a *polytope*.

Most combinatorial optimization problems can be formulated as integer programs with one-to-one correspondence between their feasible solutions are the combinatorial objects of interest. Given a graph $G = (V, E)$ and a subset of vertices $D \subseteq V$, a binary vector $x \in \{0, 1\}^{|V|}$ is a *characteristic vector* of D if $x_i = 1$ if and only if $i \in D$.

An integer programming (IP) problem in a general form is given by:

$$\max\{c^T x | Ax \leq b, \ x \in \mathbb{Z}^n\}$$

where A, b and c are as before. Denote by $Q = \{x \in \mathbb{Z}^n | Ax \leq b\}$ the set of feasible solutions to this IP, and by $P_I = \text{conv}(Q)$ – its convex hull.

Theorem 2 (see [72])**.** *A set P is a polytope if and only if there exists a finite set Q such that P is the convex hull of Q.*

Because in combinatorial optimization problems the solution sets are finite, the polyhedron considered in this dissertation is always a polytope by this theorem. If we have a complete linear description of P_I, the corresponding combinatorial optimization problem can be solved efficiently via LP approaches. However, finding the complete linear description of P_I is very difficult. Instead, we can consider the LP relaxation polyhedron P_L of the integer program above. Obviously $P_I \subseteq P_L$, so maximizing over P_L provides an upper bound on the maximum over P_I. Therefore, improving the LP relaxation using additional constraints that do not cut off any feasible integer solutions can help approaching the maximum of P_I.

Now the concepts related to strengthening the LP relaxation are introduced. A linear inequality $\pi x \leq \pi_0$ is *valid* for a polyhedron P if $P \subseteq \{x | \pi x \leq \pi_0\}$. A valid inequality $\pi x \leq \pi_0$ is said to *dominate* a valid inequality $\mu x \leq \mu_0$ if there exists $u > 0$ such that $\pi \geq u\mu$ and $\pi_0 \leq u\mu_0$ and $(\pi, \pi_0) \neq (u\mu, u\mu_0)$. A valid inequality $\pi x \leq \pi_0$ is said to be *dominated* by k valid inequalities $\pi^i x \leq \pi_0^i, i = 1, \ldots, k$ if there exist $u_i > 0$, $i = 1, \ldots, k$ such that $(\sum_{i=1}^k u_i \pi^i)x \leq \sum_{i=1}^k u_i \pi_0^i$ dominates $\pi x \leq \pi_0$.

A valid inequality restricted to be an equation represents a hyperplane. It is called a *cutting plane* or a *cut*. A polyhedron $P \subseteq \mathbb{R}^n$ is called *full-dimensional* if P contains n linearly independent directions.

Theorem 3 (see [72])**.** *If P is a full-dimensional polyhedron, it has a unique minimal description*

$$P = \{x \in \mathbb{R}^n | a^i x \leq b_i, \ i = 1, \ldots, m\}$$

where each inequality is unique to within a possible multiple.

This theorem indicates a polyhedron P can be represented by a set of cuts uniquely when P is full-dimensional.

Recall that vectors $x^1, \ldots, x^k \in \mathbb{R}^n$ are *affinely independent* if the $k-1$ directions $x^2 - x^1, \ldots, x^k - x^1$ are linearly independent. The dimension of P, denoted by $dim(P)$, is one less the maximum number of affinely independent points in P. F is a *face* of the polyhedron P if $F = \{x \in P | \pi x = \pi_0\}$ for some valid inequality $\pi x \leq \pi_0$ of P. Meanwhile $\pi x \leq \pi_0$ is said to *represent* or *define* the face F. F is a *facet* of P if F is a face and $dim(F) = dim(P) - 1$. At this time the corresponding inequality $\pi x \leq \pi_0$ which defines F is called a facet-defining inequality.

Theorem 4 (see [72])**.** *If P is full-dimensional, a valid inequality $\pi x \leq \pi_0$ is necessary in the* *description of P if and only if it defines a facet of P.*

Essentially this theorem indicates a full-dimensional polyhedron is full described by its facets and facets are the strongest among the valid inequalities. Thus facets play a crucial role in analysis of a polyhedron.

It should be noted that the number of facet-defining inequalities can be exponentially larger than the size of the original combinatorial problem. Hence, for a polytime solvable problem, it is not wise to take a cutting plane approach. This issue can be dealt with using the *ellipsoid method* [61] utilizing the the concepts of *separation* and *optimization*. A comprehensive introduction of this method is in [40]. We only present the basic ideas of separation and optimization problems. Given a rational polytope $P \subseteq \mathbb{R}^n$ and a rational vector $v \in \mathbb{R}^n$, the separation separation asks to either conclude that $v \in P$, or find a valid inequality $\pi x \leq \pi_0$ of P but violated by v. Given a rational polytope $P \subseteq \mathbb{R}^n$ and a rational vector $c \in \mathbb{R}^n$, the optimization problem asks to either find $x^* \in P$ that maximize $c^T x$ over all $x \in P$, or conclude that $P = \emptyset$.

Theorem 5 ([27])**.** *For any proper class of polyhedron, the optimization problem is polytime solvable if and only if the corresponding separation problem is polytime solvable.*

This theorem indicates that the complexity in an optimization problem is equvalent to the complxity of corresponding separation problem, not the number of facets. Therefore, if a polynomial time algorithm exists to solve a combinatorial problem, cutting-plane approaches can be applies to find one.

2.4 Convex Sets and Functions

In this section, we are focused on necessary information about convex sets and convex functions in order to formally define the concept of *curvilinear polygon*. A comprehensive reference on convex functions and convex optimization is [11]. A good reference on convex sets is [65].

First, we introduce definitions concerning convex sets. A *path* between x and y in $\mathbb{R}^n$ is a continuous function $\tau : [0, 1] \to \mathbb{R}^n$ such that $\tau(0) = x$ and $\tau(1) = y$. A set in $\mathbb{R}^n$ is said to be *(path)-connected* if for each $x_1, x_2 \in S$, there exists a path between x_1 and x_2. The definition of path-connectedness can be used as the definition of connectedness because the concepts of path-connectedness and connectedness are the same in $\mathbb{R}^n$ [65].

A set $S \subseteq \mathbb{R}^n$ is said to be *convex* if for each $x_1, x_2 \in S$, the line segment $\lambda x_1 + (1 - \lambda)x_2$ for $\lambda \in (0, 1)$ belongs to S. By the definition, every convex set is connected.

Given $x \in \mathbb{R}^n$, an *open neighborhood* of x is an open ball of positive radius centered at x; similarly a *closed neighborhood* of x is a closed ball of positive radius centered at x .

Given a set S and a point $x \in S$, x is called an *interior point* of x if there exists a open neighborhood of x that is also a subset of S. The set of interior points of S is denoted by $int(S)$. Denote $\partial S = S \setminus int(S)$ the boundary of S.

A set $S \subseteq \mathbb{R}^n$ is called *open* if $S = int(S)$. A set $S \subseteq \mathbb{R}^n$ is called *closed* if $\mathbb{R}^n \setminus S$ is open. The *closure* of set S is the intersection of all the closed sets containing S and denoted by $cl(S)$.

Proposition 1 (see [65])**.** *If S is a convex set, then $cl(S)$ is also a convex set.*

A set S is said to be *locally convex* if for every $x \in S$, there exists a closed neighborhood of x whose intersection with S is convex.

Theorem 6 (see [65])**.** *A closed connected set $S \subseteq \mathbb{R}^n$ is convex if and only if it is locally convex.*

A *supporting hyperplane* of a set $S \subseteq \mathbb{R}^n$ is a hyperplane that S is entirely contained in one of the two closed half-spaces bounded by the hyperplane and S has at least one point on the hyperplane. Let $S \subseteq \mathbb{R}^n$ and $x \in \partial S$, then S is *weakly supported at x locally* if there exists a closed neighhood $N(x)$ of x and a linear function $f \neq 0$ such that the following holds:

$$\text{if } \ y \in N(x) \text{ and } f(y) > f(x), \ \text{ then } y \notin S.$$

Proposition 2 (see [65])**.** *Let $S \subseteq \mathbb{R}^n$ and $x \in \partial S$, if there exists a closed neighborhood $N(x)$ of x whose intersection with S is convex, then S is weakly supported at x locally.*

Theorem 7 (Tietze's Theorem, see [65])**.** *Let S be an open connected subset of $\mathbb{R}^n$, then S is convex if and only if S is weakly supported locally at each of its boundary points.*

A *Jordan arc* γ in $\mathbb{R}^2$ is the image of a injective continuous map $\phi : [0,1] \to \mathbb{R}^2$. Further if ϕ is differentiable on $(0,1)$ and the derivative of ϕ is continuous, then γ is *smooth*.

Every smooth Jordan arc γ could be written as $\gamma = \mathbf{r}(t) = (x(t), y(t)), t \in [0,1]$ while both $x(t)$ and $y(t)$ are continuously differentiable functions of t on $(0,1)$. Further, the right derivatives of $x(t)$ and $y(t)$ exist on $t = 0$ and the left derivatives of $x(t)$ and $y(t)$ exist on $t = 1$. For details, see [91].

The *tangent line* to a smooth curve γ at the point $P = \mathbf{r}(t) \in \gamma$ is the straight line through the point P in the direction of vector $\mathbf{r}'(t)$.

A *convex curve* is a Jordan arc that is a connected boundary component of a convex set in $\mathbb{R}^2$.

Proposition 3 (see [91])**.** *Every point on a convex curve γ has a supporting line (supporting hyperplane in $\mathbb{R}^2$). Furthermore, if γ is smooth, then it has tangent line and the tangent line is always a supporting line.*

3. FACETS OF CONNECTED SUBGRAPH POLYTOPE

In this chapter, we introduce the (vertex-) connected subgraph polytope $\mathscr{P}(G)$ and utilize lifting technique to develop a framework to generate a large class of its facets. We also perform a study of computational complexity of this framework in different contexts.

This chapter is organized as follows. We begin in Section 3.1 with the related work about polyhedral study of connectivity and connected subgraphs and introduce the polytope $\mathscr{P}(G)$ we study with. We then discuss fundamental properties of $\mathscr{P}(G)$ in Section 3.2. In Section 3.2 we start with trivial facts of $\mathscr{P}(G)$ and then introduce lifting technique to generate non-trivial facets. As an example, we provide necessary and sufficient conditions for vertex separator inequalities to induce facets. Finally, in Section 3.3, we study the computational complexity of lifting procedure. We show that it is $\mathcal{NP}$-hard in general but polytime solvable for special graphs. We also provide insights concerning the relationship between the complexity of lifting procedure and the maximum weight connected subgraph (MWCS) problem.

3.1 Motivation

As discussed in the introduction, a polyhedral study of connected subgraph polytope is critical to study the connectivity. However, there is only a little research focused on this topic. A possible reason is that the connectivity does not have heredity property, which means the subgraph of a connected graph is not necessarily be connected. Heredity property is essential in polyhedral study of cliques, independent sets [31, 63] and clique relaxations [7] as it is essential to build *independence systems* [15].

Grötschel, Monma and Stoer [43, 41, 44] started the study of polyhedral structure of connectivity constraints arising in network design problems. Their focus was on derivation of tight inequalities of edge-connected subgraph polytope and application of these inequalities in the cutting plane method. Baïou and Mahjoub [69, 4] generalized their methods and described 2-edge connected subgraph polytope and Steiner 2-edge connected subgraph on series-parallel graphs. Barahona and Mahjoub [9] further proposed a full description of those polytopes on Halin graphs. Chopra [25] and Biha et al. [14] showed structures of k-edge connected subgraph polytope on special types of graphs. Baïou et al. [3] provided a polyhedral study on partition inequalities that are closely related to connectivity. Biha et al. [13] investigated the structure of edge connected subgraph polytope. In all these studies, the authors considered the edge connected subgraph polytope

$$eCSP(G) := \operatorname{conv}\left(\left\{x^F \in \{0,1\}^{|E|} \mid G[F] \text{ is connected}\right\}\right),$$

where x^F denotes the characteristic vector of $F \subseteq E$.

However, to the best of our knowledge, a similar but more general (as we will show later) polytope, *(vertex) connected subgraph polytope*, has not been considered in the literature.

Definition 1. *The vertex connected subgraph polytope (***CSP***) of a graph $G = (V, E)$ is*

$$\mathscr{P}(G) := \operatorname{conv}\left(\left\{x^S \in \{0,1\}^{|V|} \mid G[S] \text{ is connected}\right\}\right),$$

where x^S denotes the characteristic vector of $S \subseteq V$.

For convenience, we assume the trivial graph $(\{v\}, \emptyset)$ and the null graph $(\emptyset, \emptyset)$ are connected, thus allowing single-vertex and zero-vertex solutions. This ensures an important property that every graph's connected subgraph polytope is full-dimensional.

An integer programming formulation for $\mathscr{P}(G)$ can be obtained by enforcing the usual $0-1$ constraints and also all vertex separator inequalities, i.e., inequalities of the type

$$(a, b\text{-separator inequality}) \qquad x_a + x_b - \sum_{i \in C} x_i \leq 1,$$

where a and b are nonadjacent vertices and C is an a, b-separator. Recall that an a, b-separator C is a vertex subset (containing neither a nor b) such that nonadjacent vertices a and b are disconnected in $G[V \setminus C]$. Whenever a and b lie in different connected components of G, the empty set is an a, b-separator. Since these vertex separator inequalities are all valid, a natural question to ask is when a separator inequality is tight, i.e., when such inequality defines a facet.

A closely related problem to $\mathscr{P}(G)$ is the *maximum-weight connected subgraph* (MWCS) problem. The MWCS problem is first considered by Kerivin and Ng [60] where weight are only set on edges. They showed that version of MWCS and the prize-collecting Steiner Tree problem as defined in Johnson et al. [55] and in Geomans and Williamson [39] are equivalent optimization problems. So that problem is $\mathcal{NP}$-hard, even when restricted to planar graphs of maximum degree three with all weights either 1 or -1 [54, 93]. Using a similar approach as Feigenbaum et al. [32], that version of MWCS is shown to be $\mathcal{NP}$-hard to approximate within a constant factor.

A integer programming approach to general MWCS where weights are set in both vertices and edges can be found in [2]. It is helpful to note that by adding one vertex in the middle of each edge, general MWCS can be reformulated as the version where weights are only set on vertices.

Problem: Maximum-Weight Connected Subgraph (MWCS).

Input: a graph $G = (V, E)$, a weight w_v (possibly negative) for each $v \in V$.

Output: a maximum-weight subset $S \subseteq V$ such that $G[S]$ is connected.

One of major reasons for studying $\mathscr{P}(G)$ and $eCSP(G)$ (as in [13]) is the fact that the maximum weight connected subgraph problem can be formulated as the *linear program*

$$\begin{aligned} &\max\ w^T x \\ &\text{s.t. } x \in \mathscr{P}(G) \textit{ or } eCSP(G). \end{aligned}$$

We view $\mathscr{P}(G)$ to be more general than $eCSP(G)$ because study of $\mathscr{P}(G)$ corresponds to general MWCS while study of $eCSP(G)$ corresponds to MWCS where weights are set to edges only.

In general it is not practical to write out a full description of $\mathscr{P}(G)$. However, there is a close correspondence between the complexity of MWCS and the complexity of $\mathscr{P}(G)$. Namely, if a class of MWCS instances is difficult, we can expect the corresponding polytopes $\mathscr{P}(G)$ to be rather complex as well. Put differently, if we are seeking nice descriptions of $\mathscr{P}(G)$, we should probably restrict ourselves to easy cases of MWCS. For example, a MWCS instance on a tree is solvable in polytime, so we can expect that $\mathscr{P}(G)$ where G is a tree has rather simple structure. Our another focus is on G with a fixed independence number $\alpha(G)$ because we show MWCS is polytime solvable if $\alpha(G)$ is bounded.

3.2 Fundamental Properties of $\mathscr{P}(G)$

In this section, we describe fundamental properties of $\mathscr{P}(G)$, including when the 0-1 bounds and separator inequalities induce facets. Lifting arguments are the primary tool used in generation of facets, so we also provide some background information about lifting.

3.2.1 Trivial Facets

Proposition 4 (full-dimension; 0-1 facets)**.** *The connected subgraph polytope $\mathscr{P}(G)$ of graph G is full-dimensional. Moreover, for each $i \in V(G)$,*

1. *$x_i \geq 0$ induces a facet, and*
2. *$x_i \leq 1$ induces a facet if and only if G is connected.*

Proof. The usual $n+1$ affinely independent points $\mathbf{0}$ and e_i, $i = 1, \ldots, n$ suffice to show full-dimension. The points $\mathbf{0}$ and e_j, $j \neq i$ show that $x_i \geq 0$ induces a facet. When G is connected, consider the vertices $i = v_1, v_2, \ldots, v_n$ in a depth-first traversal ordering starting from i. Then the n affinely independent points $\sum_{j=1}^{k} e_{v_j}$ for $k = 1, \ldots, n$ show that $x_i \leq 1$ induces a facet. When G is not connected, then consider a vertex j that belongs to a different component of G than i. Then the valid inequalities $x_i + x_j \leq 1$ and $-x_j \leq 0$ imply $x_i \leq 1$, meaning that $x_i \leq 1$ cannot induce a facet. □

Lemma 1. *Consider a graph $G = (V, E)$ and a valid inequality $\sum_{i \in V} \pi_i x_i \leq \pi_0$ for $\mathscr{P}(G)$. If $S \subseteq V$, then $\sum_{i \in S} \pi_i x_i \leq \pi_0$ is valid for $\mathscr{P}(G[S])$.*

Proof. Suppose that $D \subseteq S$ is connected in $G[S]$. Then, D is also connected in G, so

$$\sum_{i \in S} \pi_i x_i^D = \sum_{i \in V} \pi_i x_i^D \leq \pi_0.$$

This concludes the proof. □

The previous proposition shows that the facets of $\mathscr{P}(G)$ depend on whether G is connected. We expound upon this in the following lemma, showing that $\mathscr{P}(G)$ is determined by its components' connected subgraph polytopes.

Theorem 8. *Let $\{G_i = (V_i, E_i)\}_i$ be the (connected) components of a graph $G = (V, E)$ and consider $(\pi, \pi_0) \in \mathbb{R}^n \times \mathbb{R}$. Then the following are equivalent.*

1. *For each G_i, the inequality $\sum_{j \in V_i} \pi_j x_j \leq \pi_0$ induces a facet of $\mathscr{P}(G_i)$.*
2. *The inequality $\sum_{j \in V} \pi_j x_j \leq \pi_0$ induces a facet of $\mathscr{P}(G)$.*

Proof. Suppose that for each G_i, the inequality $\sum_{j \in V_i} \pi_j x_j \leq \pi_0$ induces a facet of $\mathscr{P}(G_i)$. Then any subset D of vertices that induces a connected subgraph of G must belong to a single component of G, say G_k. So,

$$\sum_i \sum_{j \in V_i} \pi_j x_j^D = \sum_{j \in V_k} \pi_i x_j^D \leq \pi_0,$$

and thus $\sum_i \sum_{j \in V_i} \pi_j x_j \leq \pi_0$ is valid for $\mathscr{P}(G)$. Define $n_i := \dim(\mathscr{P}(G_i)) = |V_i|$. Then because $\sum_{j \in V_i} \pi_j x_j \leq \pi_0$ is facet-defining for $\mathscr{P}(G_i)$, there exist n_i affinely independent vectors $x^{D_i^q}$, $q = 1, \ldots, n_i$ satisfying $\sum_{j \in V_i} \pi_j x_j^{D_i^q} = \pi_0$. Add an adequate number of 0's so that $x^{D_i^q} \in \mathscr{P}(G)$. Then,

$$\sum_i \sum_{j \in V_i} \pi_j x_j^{D_i^q} = \sum_{j \in V_i} \pi_j x_j^{D_i^q} = \pi_0.$$

The total number of such vectors $x^{D_i^q}$ is $\sum n_i = dim(G)$ and the vectors are affinely independent, so $\sum_i \sum_{j \in V_i} \pi_j x_j \leq \pi_0$ is facet-defining for $\mathscr{P}(G)$.

If $\sum_i \sum_{j \in V_i} \pi_j x_j \leq \pi_0$ is facet-defining for $\mathscr{P}(G)$, then obviously $\sum_{j \in V_i} \pi_j x_j \leq \pi_0$ is valid for $\mathscr{P}(G_i)$. Since it induces a facet of $\mathscr{P}(G)$ there is a set of n affinely independent vectors $x^{D_1}, \ldots, x^{D_n} \in \mathscr{P}(G)$, each satisfying $\sum_i \sum_{j \in V_i} \pi_j x_j^{D_i} = \pi_0$. Since each of the vertex sets $D_1, \ldots, D_n$ must belong to a single component of G, for each D_q, there is a V_i such that $\sum_{j \in V_i} \pi_j x_j^{D_q} = \pi_0$. It can be argued that $N_i := \{q \mid D_q \subseteq V_i\}$ has cardinality $|V_i|$ and that the vectors x^{D_q}, $q \in N_i$ are affinely independent, implying that $\sum_{j \in V_i} \pi_j x_j \leq \pi_0$ is facet-defining for $\mathscr{P}(G_i)$. $\square$

Corollary 1. *Let $\{G_i = (V_i, E_i)\}_i$ be the components of a graph G. Then, for any $U \subseteq V$ such that $|U \cap V_i| = 1$ for each V_i, the inequality $\sum_{j \in U} x_j \leq 1$ induces a facet of $\mathscr{P}(G)$.*

Proof. By Proposition 4, for any $j \in U \cap V_i$, $x_j \leq 1$ is facet-defining for $G[V_i]$. Then by Theorem 8, $\sum_{j \in U} x_j \leq 1$ is facet-defining for $\mathscr{P}(G)$. $\square$

Corollary 2. *For a graph $G = (V, E)$ and independent set $S \subseteq V$, the inequality $\sum_{j \in S} x_j \leq 1$ induces a facet of $\mathscr{P}(G[S])$.*

While studying how inequalities define facets of $\mathscr{P}(G)$, we also need to understand when inequalities do not define facets of $\mathscr{P}(G)$. We use the following lemma throughout this dissertation to prove a inequality is not facet-defining. It is rather simple, but since we use it so often, we state it explicitly.

Lemma 2. *If $ax \leq b$ and $cx \leq d$ are valid inequalities for a* full-dimensional *polyhedron P, where (a, b) is* not *a scalar multiple of (c, d), then the inequality $(a + b)x \leq (c + d)$ is not facet-inducing for P.*

3.2.2 Basics of Lifting

Corollary 2 shows that we can easily generate the facet-defining inequality $\sum_{i \in S} x_i \leq 1$ for $\mathscr{P}(G[S])$, where S is an independent set. However, we want facet-defining inequalities for $\mathscr{P}(G)$ and this inequality is perhaps not even valid for $\mathscr{P}(G)$. *Lifting* is the procedure whereby this or other *seed* inequalities are altered so that they induce facets of $\mathscr{P}(G)$.

Lifting was first introduced by Balas [5] as a computational tool to solve integer programming problems with binary constraints. The idea of lifting is to consider the integer programming problem not in the original space, but in some space of lower dimension by enforcing certain variables to zero at the beginning. Systematic lifting procedure can be utilized to obtain strong valid inequalities and facets of polyhedra [74, 71, 43, 41, 44]. There are a variety of lifting principles, each is utilized in different fields. The lifting principle we apply is as below.

Proposition 5 (Lifting zero-valued variables, Prop. 1.1 on pp. 261 of [72])**.** *Suppose that $F \subseteq \{0, 1\}^n$, $F^\delta = F \cap \{x \in \{0, 1\}^n \mid x_1 = \delta\}$ for $\delta \in \{0, 1\}$, and $\sum_{j=2}^n \pi_j x_j \leq \pi_0$ induces a facet of* $\mathrm{conv}(F^0)$*. If $F^1 \neq \emptyset$, then*

$$(\pi_0 - \zeta)x_1 + \sum_{j=2}^{n} x_j \leq \pi_0 \tag{3.1}$$

induces a facet of $\mathrm{conv}(F)$*, where $\zeta = \max\{\sum_{j=2}^n \pi_j x_j \mid x \in F^1\}$.*

We can rewrite this lifting proposition specifically in terms of the connected subgraph polytope. It is somewhat simplified since our lifting problem is always feasible.

Corollary 3 (Lifting for $\mathscr{P}(G)$)**.** *Suppose the inequality $\sum_{j \in V \setminus \{v\}} \pi_j x_j \leq \pi_0$ induces a facet of $\mathscr{P}(G - v)$, then the inequality*

$$(\pi_0 - \zeta)x_v + \sum_{j \in V \setminus \{v\}} \pi_j x_j \leq \pi_0$$

induces a facet of $\mathscr{P}(G)$, where

$$\zeta := \max_{S \subseteq V} \left\{ \sum_{j \in V \setminus \{v\}} \pi_j x_j^S \;\middle|\; x_v^S = 1 \text{ and } G[S] \text{ is connected} \right\}.$$

This lifting principle provides a way to generate facets for $\mathscr{P}(G)$ from facets of its subgraphs' polytopes. A key idea is that this can be applied sequentially based on some lifting order. This machinery is vital for our proofs.

3.2.3 Vertex Separator Facets

We provide a good characterization for when the separator inequalities induce facets.

Theorem 9 (a,b-separator facets)**.** *Consider a connected graph $G = (V, E)$; distinct, nonadjacent vertices a and b; and a vertex subset $C \subseteq V \setminus \{a, b\}$. Then, the inequality*

$$x_a + x_b - \sum_{j \in C} x_j \leq 1$$

induces a facet of $\mathscr{P}(G)$ if and only if C is a minimal a,b-separator.

Proof. ($\implies$) Suppose that C is not an a,b-separator. Then there exists a path from a to b in $G[V \setminus C]$. Let P be the set of vertices in the path (including a and b). Then $G[P]$ is connected, but

$$x_a^P + x_b^P - \sum_{j \in C} x_j^P = x_a^P + x_b^P = 2 > 1,$$

so $x_a + x_b - \sum_{j \in C} x_j \leq 1$ is not valid. This shows that C is an a,b-separator. Now suppose C is not a *minimal* a,b-separator. Then there exists $k \in C$ such that $C \setminus \{k\}$ is an a,b-separator. Then the two valid inequalities $-x_k \leq 0$ and $x_a + x_b - \sum_{j \in C \setminus \{k\}} x_j \leq 1$ imply $x_a + x_b - \sum_{j \in C} x_j \leq 1$, so the last cannot induce a facet. This shows that C is a minimal a,b-separator.

($\impliedby$) Suppose that C is a minimal a,b-separator and define

$$A := \{v \in V \mid v \text{ and } a \text{ belong to the same component of } G[V \setminus C]\}$$

$$B := \{v \in V \mid v \text{ and } b \text{ belong to the same component of } G[V \setminus C]\}$$

$$D := V \setminus (A \cup B \cup C)$$

Claim 1: $x_a + x_b \leq 1$ induces a facet of $\mathscr{P}(G[A \cup B])$. Because C is an a,b-separator, A and B are disjoint. Further, each of $G[A]$ and $G[B]$ is connected, so by Corollary 1, $x_a + x_b \leq 1$ is facet-defining for $\mathscr{P}(G[A \cup B])$.

Claim 2: $x_a + x_b - \sum_{j \in C} x_j \leq 1$ induces a facet of $\mathscr{P}(G[A \cup B \cup C])$. Suppose $C = \{v_1, \ldots, v_c\}$ and let $C_0 := \emptyset$, $C_i := \{v_1, \ldots, v_i\}$, and $G_i := G[A \cup B \cup C_i]$. We use induction to show that, for $i = 0, 1, \ldots, c$, the inequality $x_a + x_b - \sum_{j \in C_i} x_j \leq 1$ induces a facet of $\mathscr{P}(G_i)$. When $i = 0$, the statement is true as above. Assume the statement holds when $i = k - 1$. Now consider $i = k$. By the induction assumption, $x_a + x_b - \sum_{j \in C_{k-1}} x_j \leq 1$ is facet-defining for $\mathscr{P}(G_{k-1})$. Define

$$\zeta := \max_{S \subseteq A \cup B \cup C_k} \left\{ x_a^S + x_b^S - \sum_{j \in C_{k-1}} x_j^S \;\middle|\; x_{v_k}^S = 1 \text{ and } G[S] \text{ is connected} \right\}.$$

On one hand,

$$x_a + x_b - \sum_{j \in C_{k-1}} x_j \leq x_a + x_b \leq 2,$$

so $\zeta \leq 2$. Also, because C is a minimal separator of a and b, there is a path from a to b in $G[(V \setminus C) \cup \{v_k\})]$, let T be the set of vertices in this path, then x^T is feasible for the lifting problem, and

$$x_a^T + x_b^T - \sum_{j \in C_{k-1}} x_j^T = x_a^T + x_b^T = 2,$$

so $\zeta \geq 2$. This implies that $\zeta = 2$, and by the lifting principle, the inequality

$$(1 - \zeta) x_{v_k} + x_a + x_b - \sum_{j \in C_{k-1}} x_j = x_a + x_b - \sum_{j \in C_k} x_j \leq 1$$

induces a facet of $\mathscr{P}(G_k)$, so the statement is true when $i = k$ and in general. Thus, $x_a + x_b - \sum_{j \in C} x_j \leq 1$ is facet-defining for $\mathscr{P}(G_c) = \mathscr{P}(G[A \cup B \cup C])$.

Claim 3: $x_a + x_b - \sum_{j \in C} x_j \leq 1$ induces a facet of $\mathscr{P}(G)$. For any $u \in D$, let $\sigma(u)$ be the length of a shortest path from u to (a vertex of) C measured in terms of the number of edges. Note that C is nonempty and G is connected, so $\sigma(u)$ is well-defined, i.e., $0 < \sigma(u) < \infty, \forall u \in D$. Order $D = \{u_1, \ldots, u_d\}$ such that $\sigma(u_s) \leq \sigma(u_t), \forall s \leq t$, e.g., by breadth-first search. Let $D_0 = \emptyset$, $D_i = \{u_1, \ldots, u_i\}$, and $H_i = G[(V \setminus D) \cup D_i]$. We use induction to show that, for $i = 0, 1, \ldots, d$, the inequality $x_a + x_b - \sum_{j \in C} x_j \leq 1$ induces a facet of $\mathscr{P}(H_i)$. When $i = 0$, we already know that $x_a + x_b - \sum_{j \in C} x_j \leq 1$ is facet-defining for $\mathscr{P}(H_0)$. So, assume the statement holds when $i = k-1$, and consider $i = k$. By the induction assumption, $x_a + x_b - \sum_{j \in C} x_j \leq 1$ is facet-defining for $\mathscr{P}(H_{k-1})$. Define

$$\zeta := \max_{S \subseteq V(H_k)} \left\{ x_a^S + x_b^S - \sum_{j \in C} x_j^S \;\middle|\; x_{u_k}^S = 1 \text{ and } G[S] \text{ is connected} \right\}.$$

Consider a feasible solution $S \subseteq V(H_k)$ to the lifting problem. On one hand, if $x_a^S + x_b^S - \sum_{j \in C} x_j^S > 1$, then both a and b belong to S. But, for $G[S]$ to be connected, there must exist $q \in C \cap S$. So

$$x_a^S + x_b^S - \sum_{j \in C} x_j^S \leq x_a^S + x_b^S - x_q^S = 1,$$

which is a contradiction. This shows $\zeta \leq 1$. Now we show the reverse inequality. Let T_1 be the set of vertices in a shortest path from u_k to C in G and suppose $q \in C$ is the other end point in the path. Then $T_1 \cap C = \{q\}$, since otherwise T_1 is not a shortest path. Further, since C is a minimal a,b-separator, there is a path from a to b in $G[(V \setminus C) \cup \{q\}]$. Let T_2 be the set of vertices in this path, and let $T = T_1 \cup T_2$. Then $G[T]$ is connected, $a, b, u_k, q \in T$ and $T \cap C = \{q\}$, so x^T is feasible for the lifting problem and

$$x_a^T + x_b^T - \sum_{j \in C} x_j^T = x_a^T + x_b^T - x_q^T = 1,$$

so $\zeta \geq 1$. Thus $\zeta = 1$, and by the lifting principle, the inequality $x_a + x_b - \sum_{j \in C} x_j \leq 1$ induces a facet of $\mathscr{P}(H_k)$. So, the statement is true when $i = k$ and is true in general. Thus $x_a + x_b - \sum_{j \in C} x_j \leq 1$ is facet-defining for $\mathscr{P}(H_d) = \mathscr{P}(G)$. □

3.3 Complexity of Lifting

One may wonder how difficult it is to generate a facet-defining inequality for $\mathscr{P}(G)$ via lifting. In this section, we show that this problem is hard in general, but is polytime solvable in several special cases.

Lemma 3. *Consider a facet-defining inequality $\sum_{i \in V} \pi_i x_i \leq \pi_0$ of $\mathscr{P}(G)$. Then $\pi_0 \geq 0$. If the inequality is not a multiple of $-x_i \leq 0$, then $\pi_0 > 0$.*

Proof. As the empty set is connected, $\pi_0 \geq 0$. Suppose that $\pi_0 = 0$. Then $\pi_i \leq 0$ for each vertex

$i \in V$ (since the trivial graphs are connected). Further suppose that at least two coefficients are negative, say π_u and π_v. Then $\sum_{i \in V} \pi_i x_i \leq 0$ is implied by the valid inequalities $-\pi_u x_u \leq 0$ and $\sum_{i \in V \setminus \{u\}} \pi_i x_i \leq 0$. These two new inequalities are distinct, so Lemma 2 shows that $\sum_{i \in V} \pi_i x_i \leq 0$ cannot be facet-defining. $\square$

Lemma 4 (Bounds on lifting)**.** *Suppose $\sum_{i \in V \setminus \{v\}} \pi_i x_i \leq \pi_0$ is facet-defining for $\mathscr{P}(G - v)$. Then, when lifting in v, the objective ζ of the lifting problem satisfies:*

1. *if v is isolated, then $\zeta = 0$;*
2. *if v is not isolated, then $\pi_0 \leq \zeta \leq |N_G(v)|\pi_0$.*

Proof. The single-vertex solution $\{v\}$ implies that $\zeta \geq 0$. When v is isolated, the only feasible solution is $\{v\}$, in which case $\zeta = 0$. So, from now on we will suppose that $N_G(v) \neq \emptyset$.

Consider an optimal solution $D \subseteq V(G)$ to the lifting problem. Here, $v \in D$ and $G[D]$ is connected. Suppose $N_G(v) = \{u_1, \ldots, u_s\}$. Partition $D' := D \setminus \{v\}$ into s (possibly empty) subsets as follows. Let D_1 denote the set of vertices in D' connected to u_1 by some path of $G[D']$. Then for $p = 2, \ldots, s$, let D_p denote the vertices of $D' \setminus (D_1 \cup \cdots \cup D_{p-1})$ that are connected to u_p by some path in $G[D']$. Each $G[D_p]$ is a connected subgraph of $G - v$, so by the validity of the seed inequality,

$$\sum_{j \in V(G)} \pi_j x_j^{D_p} = \sum_{j \in D_p} \pi_j \leq \pi_0,$$

implying that

$$\begin{aligned}
\zeta \quad &= \sum_{j \in V(G-v)} \pi_j x_j^D \\
&= \sum_{p=1}^{s} \left(\sum_{j \in D_p} \pi_j \right) + \sum_{j \in V(G) \setminus D} \pi_j x_j^{D_p} \\
&\leq s\pi_0 + 0 \\
&= |N_G(v)|\pi_0.
\end{aligned}$$

Now suppose $N_G(v) \neq \emptyset$, and choose $u \in N_G(v)$. Let $G' = (V', E')$ be the connected component of $G - v$ that includes u. Then, by Theorem 8, $\sum_{j \in V'} \pi_j x_j \leq \pi_0$ is facet-defining for $\mathscr{P}(G')$. Moreover, there must be at least one connected vertex subset $D \subseteq V'$ containing u for which $\sum_{j \in V'} \pi_j x_j^D = \pi_0$, since otherwise the inequality could not induce a facet. Then, $G[D + v]$ is connected and has weight π_0, so $\zeta \geq \pi_0$. $\square$

3.3.1 Lifting in Bipartite Subgraphs

Here we show that lifting arguments find a nontrivial closed-form facet for bipartite subgraphs. However, for a different lifting order, the lifting problem is $\mathcal{NP}$-hard.

The easy and hard lifting orders are only slightly different, and we find the stark contrast in complexity very interesting. Let the vertex partitions of the bipartite graph be A and B, where $A = \{a_1, \ldots, a_p\}$ and $B = \{b_1, \ldots, b_q\}$. When the lifting order is

$$a_1, \ldots, a_p, b_1, \ldots, b_q$$

then the resulting facet can be generated in linear time. In fact, changing the order of vertices within A (or within B) does not change the facet. However, the slightly different lifting order

$$a_2, \ldots, a_p, b_1, \ldots, b_q, a_1$$

results in the $\mathcal{NP}$-hard problem of lifting in vertex a_1.

Theorem 10 (Bipartite lifting–easy case)**.** *For a bipartite graph $G = (V, E)$ with independent partitions $A \subseteq V$ and $B = V \setminus A$, the following inequality induces a facet of $\mathscr{P}(G)$.*

$$\sum_{j \in A} x_j - \sum_{j \in B} (|N_G(j)| - 1) x_j \leq 1.$$

Proof. Suppose $B = \{v_1, \ldots, v_b\}$ and let $B_0 = \emptyset$, $B_i = \{v_1, \ldots, v_i\}$, and $G_i = G[A \cup B_i]$. We use induction to prove that

$$\sum_{j \in A} x_j - \sum_{j \in B_i} (|N_G(j)| - 1) x_j \leq 1$$

induces a facet of $\mathscr{P}(G_i)$.

When $i = 0$, the inequality $\sum_{j \in A} x_j \leq 1$ induces a facet of $\mathscr{P}(G_0) = \mathscr{P}(G[A])$, which follows by Corollary 2. Suppose the statement is true for $i = k-1$, and consider $i = k$. To apply the lifting principle, define

$$\zeta = \max_{S \subseteq A \cup B_k} \left\{ \sum_{j \in A} x_j^S - \sum_{j \in B_{k-1}} (|N_G(j)| - 1) x_j^S \;\middle|\; x_k^S = 1 \text{ and } G[S] \text{ is connected} \right\}.$$

First see that $\zeta \geq |N_G(v_k)|$, since vertex v_k along with its neighborhood (a subset of A) provides a solution with $x_k = 1$ of weight $|N_G(v_k)|$. The reverse inequality $\zeta \leq |N_G(v_i)|$ follows by Lemma 4. So, $\zeta = |N_G(v_k)|$, and by the lifting principle, $\sum_{j \in A} x_j - \sum_{j \in B_i}(|N_G(v_j)| - 1)x_j \leq 1$ is facet-defining for $\mathscr{P}(G_i)$, so the statement is true when $i = k$. Thus the statement is true in general, and

$$\sum_{j \in A} x_j - \sum_{j \in B}(|N_G(v_j)| - 1)x_j \leq 1$$

is facet-defining for $\mathscr{P}(G) = \mathscr{P}(G_b)$. □

Theorem 11 (Bipartite lifting–hard case)**.** *Lifting a vertex v into a given facet-defining inequality of $\mathscr{P}(G - v)$ is $\mathcal{NP}$-hard, even when graph G is bipartite and 2-degenerate.*

Proof. The reduction is from 3OCC-3SAT, a special case of 3SAT in which each variable appears at most three times and each literal appears at most twice. This remains $\mathcal{NP}$-complete; cf. Theorem 16.5 of [75]. Let the instance $\Phi = \bigwedge_{j=1}^{m}(c_j^1 \vee c_j^2 \vee c_j^3)$ of 3OCC-3SAT be defined over variables $x_1, \ldots, x_n$. We construct a graph $G = (V, E)$ and a lifting order for which the *final* lifting problem has objective $2n + m$ if and only Φ is satisfiable. In contrast, a linear-time algorithm computes all other lifted coefficients.

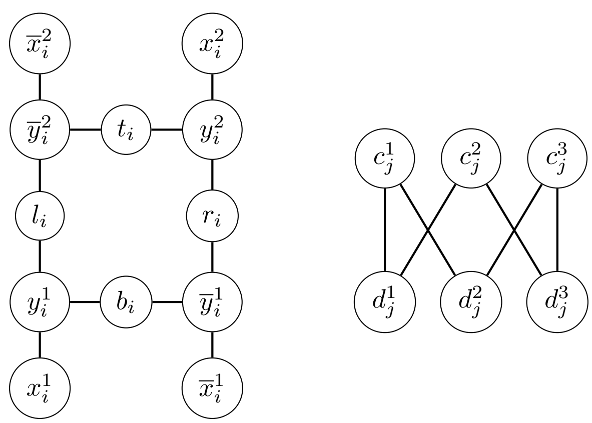


Figure 3.1: Variable gadget (left) and clause gadget (right).

For each variable x_i and for each clause c_j in the 3-OCC-3SAT instance, construct a gadget, as shown in Figure 3.1. Connect the gadgets as follows. Connect each literal $x_i(\overline{x}_i)$ from a clause

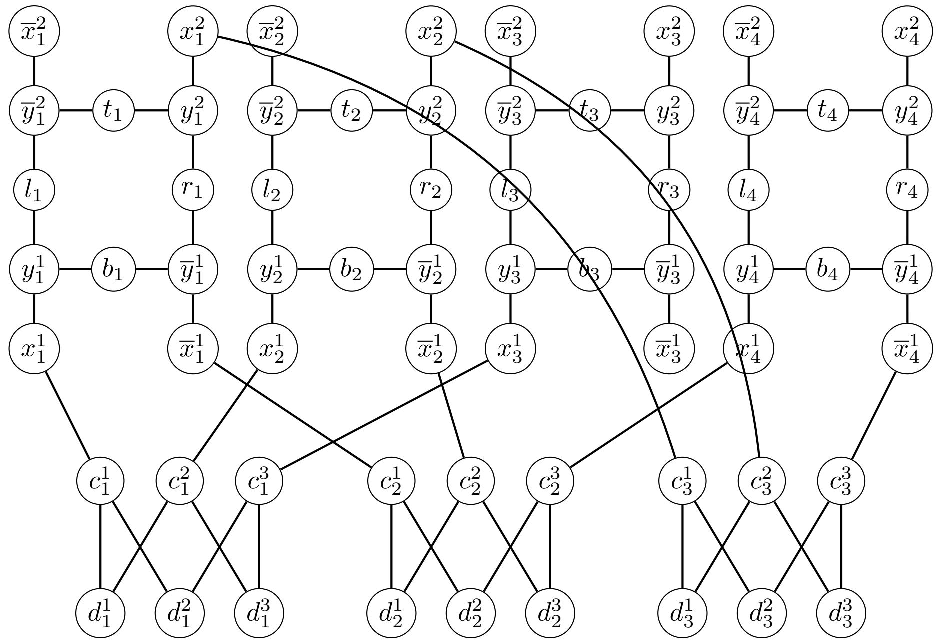


Figure 3.2: The construction of graph $G - v$ when given 3OCC-3SAT instance $\Phi = (x_1 \vee x_2 \vee x_3) \wedge (\overline{x}_1 \vee \overline{x}_2 \vee x_4) \wedge (x_1 \vee v_2 \vee \overline{x}_4)$.

gadget to a literal x_i^1 or x_i^2 ($\overline{x}_i^1$ or $\overline{x}_i^2$) from the corresponding variable gadget. Because each literal appears in at most two clauses, we can suppose that no pair of clause vertices are connected to the same variable gadget literal. This is illustrated in Figure 3.2. Finally, add a new vertex v and connect it to every clause vertex of the type c_j^k and to all vertices of the type y_i^1, y_i^2, $\overline{y}_i^1$ and $\overline{y}_i^2$. Since the number of vertices of G is only $12n + 6m + 1$ the reduction is polynomial.

First see that G is bipartite, with partitions A and B:

$$A = \{v\} \cup \left(\bigcup_{i=1}^{n} \{l_i, r_i, b_i, t_i, x_i^1, x_i^2, \overline{x}_i^1, \overline{x}_i^2\} \right) \cup \left(\bigcup_{j=1}^{m} \{d_j^1, d_j^2, d_j^3\} \right);$$

$$B = \left(\bigcup_{i=1}^{n} \{y_i^1, y_i^2, \overline{y}_i^1, \overline{y}_i^2\} \right) \cup \left(\bigcup_{j=1}^{m} \{c_j^1, c_j^2, c_j^3\} \right).$$

Now we show G is 2-degenerate. Suppose not; then there is a subgraph H of G in which all vertices have degree at least three. Then H cannot contain a vertex of the type d_j^k, l_i, r_i, t_i, or b_i, since these vertices have degree at most two in G. Now, if those vertices do not belong to H, then H cannot contain a vertex of the type $y_i^1, y_i^2, \overline{y}_i^1, \overline{y}_i^2$, or c_j^k. This implies that $V(H) \subseteq$

$\{v\} \cup \left(\bigcup_{i=1}^{n} \{x_i^1, x_i^2 \overline{x}_i^1, \overline{x}_i^2\}\right)$, meaning that $V(H)$ is independent, but this contradicts that H has minimum degree at least three.

Since $G' = G - v$ is bipartite with partitions $A \setminus \{v\}$ and B, Theorem 10 implies that the following inequality induces a facet of $\mathscr{P}(G')$.

$$\sum_{j \in A \setminus \{v\}} x_j - \sum_{j \in B} 2x_j \leq 1 \tag{3.2}$$

Now, consider the problem of lifting v into inequality (3.2), i.e., solving for

$$\zeta := \max_{S \subseteq V} \left\{ \sum_{j \in A \setminus \{v\}} x_j^S - \sum_{j \in B} 2x_j^S \;\middle|\; x_v^S = 1 \text{ and } G[S] \text{ is connected} \right\}.$$

Claim 1: There is an optimal solution $D \subseteq V$ to the lifting problem that satisfies:

- for each i, either $\{y_i^1, y_i^2\} \subseteq D$ or $\{\overline{y}_i^1, \overline{y}_i^2\} \subseteq D$, but not both; and
- for each j, exactly one of c_j^1, c_j^2, and c_j^3 belongs to D.

If an optimal solution $D \subseteq V$ to the lifting problem does not fit these criteria, it can be modified so that does. Recognize that $v \in D$ and consider the following cases.

1. Three or more of $\{y_i^1, y_i^2, \overline{y}_i^1, \overline{y}_i^1\}$ belong to D. Without loss of generality, suppose that $\{y_i^2, \overline{y}_i^1, \overline{y}_i^1\} \subseteq D$, thus we can assume that $x_i^2 \in D$. Then $D' = D \setminus \{y_i^2, x_i^2\}$ is connected, contains v, and has a larger weight than D, a contradiction.

2. Two of $\{y_i^1, y_i^2, \overline{y}_i^1, \overline{y}_i^1\}$ belong to D. If either $\{y_i^1, y_i^2\} \subseteq D$ or $\{\overline{y}_i^1, \overline{y}_i^2\} \subseteq D$, then Claim 1 is satisfied. Otherwise, without loss of generality, suppose that y_i^1 and $\overline{y}_i^1$ belong to D. Then t_i cannot belong to D by connectivity. We can assume that x_i^1 belongs to D. Now, $D' = D \cup \{\overline{y}_i^2, t_i\} \setminus \{x_i^1, y_i^1\}$ is connected, contains v, and has the same weight.

3. One of $\{y_i^1, y_i^2, \overline{y}_i^1, \overline{y}_i^1\}$ belongs to D. Without loss of generality, suppose that y_i^1 belongs to D. Then $D' = D \cup \{y_i^2, t_i, r_i\}$ is connected, contains v, and has the same weight.

4. None of $\{y_i^1, y_i^2, \overline{y}_i^1, \overline{y}_i^1\}$ belong to D. Then $D' = D \cup \{y_i^1, y_i^2, t_i, b_i, l_i, r_i\}$ is connected, contains v, and has the same weight.

5. Two or more of $\{c_j^1, c_j^2, c_j^3\}$ belong to D. Without loss of generality, suppose that $c_j^1, c_j^2 \in D$. We can assume that $d_j^1, d_j^2, d_j^3 \in D$, and that c_j^1 has a neighbor, say w, from a variable gadget

and w also belongs to D. Then $D' = D \setminus \{c_j^1, d_j^2, w\}$ is connected, contains v, and has the same weight.

6. None of $\{c_j^1, c_j^2, c_j^3\}$ belong to D. Then $D' = D \cup \{c_j^1, d_j^1, d_j^2\}$ is connected, contains v, and has the same weight.

These steps can be applied repeatedly until D satsfies the claim.

Claim 2: $\zeta \leq 2n + m$. Consider an optimal solution $D \subseteq V$ to the lifting problem that satisfies Claim 1. See that any weight $+1$ vertex in D must have a weight -2 neighbor in D. There are $2n + m$ vertices of weight -2 in D and each has three weight $+1$ neighbors in G. So,

$$\begin{aligned}
\zeta &= \sum_{j \in A \setminus \{v\}} x_j^D - \sum_{j \in B} 2x_j^D \\
&= \sum_{j \in A \setminus \{v\}} x_j^D - 2(2n + m) \\
&\leq 3(2n + m) - 2(2n + m) = 2n + m.
\end{aligned}$$

Claim 3: If Φ is satisfiable, then $\zeta \geq 2n + m$. Given a satisfying assignment x^* for Φ, construct a solution D to the lifting problem as follows.

- For each i: if $x_i^* = 1$, choose $\overline{y}_i^1$ and $\overline{y}_i^1$; otherwise, select y_i^1 and y_i^2. Note that this is, in a sense, the opposite of the satisfying assignment.
- For each j: the satisfying assignment makes clause j evaluate to true by some literal, say c_j^k; choose vertex c_j^k and also the neighboring vertex from the variable gadget.
- Add v and all positive-weight vertices that neighbor a previously chosen vertex.

This solution D is feasible, since all negative-weight vertices are adjacent to v, and their positive-weight neighbors were chosen. One negative-weight vertex was chosen from each clause gadget, and two negative-weight vertices were selected from each variable gadget. So, there are $2n + m$ vertices of negative-weight in D. Each of these negative-weight vertices has three positive-weight neighbors. All that remains is to demonstrate that no two negative-weight vertices of D share a neighbor of positive weight. The proof of this is straightforward but tedious, so we omit it. Thus D has weight $(2n + m)(-2 + 3(1)) = 2n + m$.

Claim 4: If $\zeta \geq 2n+m$, then Φ is satisfiable. Consider an optimal solution $D \subseteq V$ that satisfies Claim 1 and has weight at least $2n+m$. Then,

- for each i, either $\{y_i^1, y_i^2\} \subseteq D$ or $\{\overline{y}_i^1, \overline{y}_i^2\} \subseteq D$, but not both; and
- for each j, exactly one of c_j^1, c_j^2, and c_j^3 belongs to D.

The following assignment x^* will be shown to satisfy Φ. For each i: if $\{y_i^1, y_i^2\} \subseteq D$, then set $x_i^* = 0$; otherwise, set $x_i^* = 1$. Then $\zeta = 2n+m$ by Claim 2, and this equality holds if and only if no two negative-weight vertices in D have a common neighbor (of positive weight).

We argue that, for each j, x^* makes clause j evaluate to true. Let $c_j^{k_j}$ be the vertex from clause j that belongs to D. Suppose that the neighbor of c_j^k from the variable gadget is

- x_i^δ for some $\delta \in \{1,2\}$. Then, y_i^δ does not belong to D, so $\overline{y}_i^1$ and $\overline{y}_i^2$ belong to D and thus $x_i^* = 1$, which satisfies clause j.
- $\overline{x}_i^\delta$ for some $\delta \in \{1,2\}$. Then, $\overline{y}_i^\delta$ does not belong to D, so y_i^1 and y_i^2 belong to D and thus $x_i^* = 0$, which satisfies clause j.

So x^* is a satisfying assignment.

By Claim 3 and 4, the *final* lifting problem has objective $2n+m$ if and only Φ is satisfiable. Then, since 3OCC-3SAT is $\mathcal{NP}$-hard and since the reduction is polynomial, the problem of lifting in v into inequality (3.2) is $\mathcal{NP}$-hard. □

3.3.2 Polytime Cases of Lifting

As noted previously, lifting v into a facet of $\mathscr{P}(G-v)$ is an instance of the MWCS problem. So we derive the following theorem.

Theorem 12. *Consider a connected graph $G=(V,E)$ and an independent set $S \subseteq V$ with $|S| =$ polylog(n). Then there exists an order of lifting in $V \setminus S$ such that the lifting procedure from S via that order can be done in polynomial time of n.*

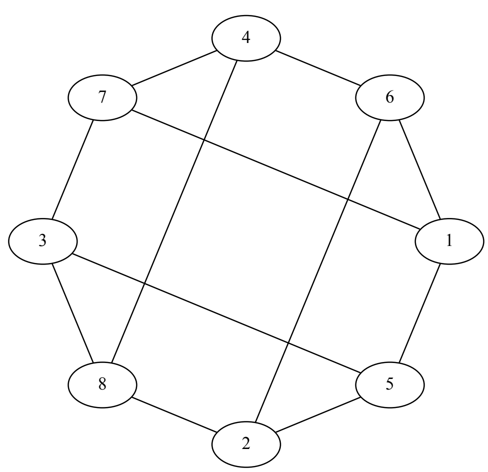
Proof. By Corollary 2, $\sum_{v_i \in S} x_i \leq 1$ is facet-defining for $\mathscr{P}(G[S])$. Arbitrarily choose $v \in S$ and order the set $V \setminus S$ as $v_1, \ldots, v_m$ in a depth-first traversal ordering starting from v. Let $D_i = S \cup \{v_1, \ldots, v_i\}$ then $N_{G[D_{i-1}]}(v_i) \neq \emptyset$, so by lemma 4, $\zeta \geq \pi_0$ and then $\pi_i \leq 0$ for any $v_i \in V \setminus S$. In every calculation of ζ, we in fact solve a maximum weight connected subgraph problem($MWCSP$) for $G[D_i]$ with weight function $w : D_{i-1} \to \mathbb{R}$ such that $w(v_i) = \pi_i$ and a terminal v_i. As $\pi_i > 0$ only for $v_i \in S$, the number of positive weight vertices in $MWCSP$ is bounded by C, so $MWCSP$ can be solved in time $\mathcal{O}(4^C poly(n))$ [94], so every calculation of ζ can be done in polynomial time of n and then the lifting procedure can also be done in polynomial time of n. $\square$

4. DESCRIPTION OF CONNECTED SUBGRAPH POLYTOPE

In this chapter, we focus on classes of graphs G where all facets of $\mathscr{P}(G)$ can be generated by lifting procedure. First, in Section 4.1, we present a counterexample to show this is not true for general G. We then investigate three classes of G where this is true: the first one is graphs whose independence number is no more than two, while in this case we in fact show vertex separator inequalities together with bound inequalities characterize $\mathscr{P}(G)$; the second one is *3-plex* and *3-defective cliques*; the third one is forests, and we further develop a linear-time algorithm to generate every facet. In Sections 4.2, 4.3 and 4.4 we discuss the three cases respectively.

4.1 Facets Not Generated by Lifting

Lifting is powerful tool to generate facets of connected subgraph polytope. However, in each step that we apply lifting principle in corollary 3, the right-side value of the inequality never changes, and then the right-side value should always be equal to the value of a positive coefficient in a facet-defining inequality because we start the lifting procedure by corollary 2. Therefore, not all facets of $\mathscr{P}(G)$ can be generated by lifting procedure for a general graph G because not all facets have this property. In fact, let G be a 3-*cube* as in the figure,



then $x_1 + x_2 + x_3 + x_4 - x_5 - x_6 - x_7 - x_8 \le 2$ is facet-defining for $\mathscr{P}(G)$ by computation, however, this facet can not be generated by lifting procedure as all positive coefficients are 1 while the right-side value is 2.

Moreover, there exist facet-defining inequalities that the right-side values can be arbitrarily large while all positive left-side coefficients are 1.

Proposition 6. *Let $G = (V, E)$ with $v = [1..2n]$ and $E = \{(i, j) | i \in [1..n], j \in [n+1, 2n], i+n \neq j\}$, then*

$$-\sum_{i=1}^{n} x_i + \sum_{j=n+1}^{2n} x_j \le n - 2$$

defines a facet of $\mathscr{P}(G)$.

Proof. For any induced connected subgraph $G[D]$, if $D \cap [1..n] = \emptyset$, because $[n+1, 2n] \subset V$ is an independent set, $|D \cap [n+1, 2n]| \le 1$ and $-\sum_{i=1}^{n} x_i^D + \sum_{j=n+1}^{2n} x_j^D \le 1$.

If $|D \cap [1..n]| = 1$, suppose $D \cap [1..n] = \{i\}$, then $i + n \notin D$, so $|D \cap [n+1..2n]| \le n - 1$, thus $-\sum_{i=1}^{n} x_i^D + \sum_{j=n+1}^{2n} x_j^D \le n-2$; if $|D \cap [1..n]| \ge 2$, $-\sum_{i=1}^{n} x_i^D + \sum_{j=n+1}^{2n} x_j^D \le n-2$. Therefore,

$$-\sum_{i=1}^{n} x_i + \sum_{j=n+1}^{2n} x_j \le n - 2$$

is valid in $\mathscr{P}(G)$.

To show that $-\sum_{i=1}^{n} x_i + \sum_{j=n+1}^{2n} x_j \le n - 2$ defines a facet in $\mathscr{P}(G)$, we need to find $2n$ affinely independent x^D such that $-\sum_{i=1}^{n} x_i^D + \sum_{j=n+1}^{2n} x_j^D = n - 2$.

In fact, for $i \in [1..n]$ let $D_i = [n+1..2n] \cap \{i\} \setminus \{n+i\}$, for $i \in [n+2, 2n]$ let $D_i = [n+1, 2n] \cap \{1, i-n\}$ and $D_{n+1} = [n+1, 2n] \cap \{2, 3\}$, obviously every $G[D_i]$ is connected and

$$-\sum_{i=1}^{n} x_i^{D_i} + \sum_{j=n+1}^{2n} x_j^{D_i} = n - 2$$

for every D_i, $i \in [1..2n]$.

Next we show $x^{D_i}, i \in [1..2n]$ are linearly independent and thus affinely independent. Let $\boldsymbol{D} = (x^{D_1}, x^{D_2}, ..., x^{D_n})^T$, then

$$\boldsymbol{D} = \begin{pmatrix} I_n & \mathbf{1} - I_n \\ S & \mathbf{1} \end{pmatrix}$$

Where $\mathbf{1}$ is a $n \times n$ matrix of ones and

$$\boldsymbol{S}^{n\times n} = \begin{pmatrix} 0 & 1 & 1 & & \\ 1 & 1 & 0 & \text{\Large 0} & \\ \vdots & & \ddots & & \\ \vdots & \text{\Large 0} & & \ddots & \\ 1 & & & & 1 \end{pmatrix}$$

Easy calculations show

$$\det \boldsymbol{D} = \det \begin{pmatrix} I_n & \mathbf{1} - I_n \\ \mathbf{0} & (S) - \mathbf{1} \end{pmatrix} > 0$$

So $x^{D_i}, i \in [1..2n]$ are linearly independent and thus

$$-\sum_{i=1}^{n} x_i + \sum_{j=n+1}^{2n} x_j \leq n - 2$$

defines a facet of $\mathscr{P}(G)$. □

4.2 When Vertex Separator Inequalities Characterize $\mathscr{P}(G)$

The 0-1 bounds and vertex separator inequalities provide a natural integer programming formulation for $\mathscr{P}(G)$. Namely, the *integer hull* of the polytope

$$Q(G) := \left\{ x \in [0,1]^{|V|} \;\middle|\; x \text{ satisfies all vertex separator inequalities} \right\}$$

is precisely $\mathscr{P}(G)$. Note that $Q(G)$ provides a tractable relaxation for $\mathscr{P}(G)$, as one can optimize over $Q(G)$ in polytime via the ellipsoid method or reformulation of $Q(G)$ by network flow constraints.

4.2.1 Reformulation of $Q(G)$

Consider a simple graph $G = (V, E)$. For convenience, let the set $\overline{E}$ of complement edges include both directions. The polytope $F(G)$ is is the set of all (x, f) satisfying the following flow network constraints.

$$-x_i + \sum_{j \in N(i)} f_{ij}^{ab} \leq 0, \ \forall ab \in \overline{E}, \ \forall i \in V \tag{4.1}$$

$$x_a + x_b - \left(\sum_{j \in N(a)} f_{aj}^{ab} - \sum_{j \in N(a)} f_{ja}^{ab} \right) \leq 1, \ \forall ab \in \overline{E} \tag{4.2}$$

$$\sum_{j \in N(i)} f_{ji}^{ab} - \sum_{j \in N(i)} f_{ij}^{ab} = 0, \ \forall i \in V \setminus \{a, b\}, \ \forall ab \in \overline{E} \tag{4.3}$$

$$0 \leq x_i \leq 1, \ \forall i \in V \tag{4.4}$$

$$0 \leq f_{ij}^{ab} \leq 1, \ \forall ij \in E, \ \forall ab \in \overline{E} \tag{4.5}$$

Lemma 5. $\text{proj}_x(F(G)) \subseteq Q(G)$.

Proof. Let $(x, f) \in F(G)$. Consider arbitrary $ab \in \overline{E}$ and an a, b-separator $C \subseteq V \setminus \{a, b\}$. Let A be the set of vertices reachable from a in $G[V \setminus C]$ and let $B = V \setminus (A \cup C)$. For convenience, define $f_{ij}^{ab} = 0$ whenever $\{i, j\} \notin E$. Then,

$$\begin{aligned}
x_a + x_b - 1 &\leq \sum_{j \in V} f_{aj}^{ab} - \sum_{j \in V} f_{ja}^{ab} && (4.6)\\
&= \sum_{i \in A \cup C} \left(\sum_{j \in V} f_{ij}^{ab} - \sum_{j \in V} f_{ji}^{ab} \right) && (4.7)\\
&= \sum_{i \in A \cup C} \sum_{j \in A \cup C} \left(f_{ij}^{ab} - f_{ji}^{ab} \right) + \sum_{i \in A \cup C} \sum_{j \in B} \left(f_{ij}^{ab} - f_{ji}^{ab} \right) && (4.8)\\
&= \sum_{i \in A \cup C} \sum_{j \in B} \left(f_{ij}^{ab} - f_{ji}^{ab} \right) && (4.9)\\
&= \sum_{i \in C} \sum_{j \in B} \left(f_{ij}^{ab} - f_{ji}^{ab} \right) && (4.10)\\
&\leq \sum_{i \in C} \sum_{j \in B} f_{ij}^{ab} && (4.11)
\end{aligned}$$

$$\begin{aligned}
&\leq \sum_{i \in C} \sum_{j \in V} f_{ij}^{ab} && (4.12)\\
&\leq \sum_{i \in C} x_i. && (4.13)
\end{aligned}$$

Thus, $x_a + x_b - \sum_{i \in C} x_i \leq 1$ and $0 \leq x \leq 1$, so $x \in Q(G)$. $\square$

Lemma 6. $Q(G) \subseteq \text{proj}_x(F(G))$.

Proof. Let $x \in Q(G)$. Consider arbitrary $ab \in \overline{E}$ and the maximization problem

$$\begin{aligned} F = \max \sum_{j \in N(a)} f_{aj}^{ab} & \\ s.t. \sum_{j \in N(i)} f_{ij}^{ab} \leq x_i,\ & \forall i \in V \\ \sum_{j \in N(i)} f_{ji}^{ab} - \sum_{j \in N(i)} f_{ij}^{ab} = 0,\ & \forall i \in V \setminus \{a, b\} \\ 0 \leq f_{ij}^{ab} \leq 1,\ & \forall ij \in E,\ \forall j \neq a \\ f_{ia}^{ab} = 0,\ & \forall ia \in E \end{aligned}$$

This is a maximum network flow problem with node capacities. Ford and Fulkerson [33, Chapter I.11] studied this problem and according to them the maximum flow value is equal to the capacity of the $a, b-$separator with minimum capacity, i.e., $F = \min_C \sum_{i \in C} x_i$ where C is a a, b-separator.

Because $x \in Q(G)$, for any $a.b$-separator C, we have $\sum_{i \in C} x_i \geq x_a + x_b - 1$, so $F \geq x_a + x_b - 1$, therefore the system

$$\sum_{j \in N(a)} f_{aj}^{ab} \geq x_a + x_b - 1$$

$$\sum_{j \in N(i)} f_{ij}^{ab} \leq x_i,\ \forall i \in V$$

$$\sum_{j \in N(i)} f_{ji}^{ab} - \sum_{j \in N(i)} f_{ij}^{ab} = 0,\ \forall i \in V \setminus \{a, b\}$$

$$0 \leq f_{ij}^{ab} \leq 1,\ \forall ij \in E,\ \forall j \neq a$$

$$f_{ia}^{ab} = 0,\ \forall ia \in E$$

is feasible and suppose f^{ab} is a feasible solution. Obviously f^{ab} satisfies (1) through (5) for the given a,b. Let f be f^{ab} though all $ab \in E$, then f satisfies (1) through (5) and thus $(x, f) \in F(G)$, so $x \in \text{proj}_x(F(G))$. $\square$

Theorem 13. $Q(G) = \text{proj}_x(F(G))$.

Proof. It is a combination of Lemma 5 and 6. $\square$

The number of constraints to define $F(G)$ is bounded by $\mathcal{O}(|V|^4)$, so linear optimization problem over $F(G)$ in polytime solvable.

4.2.2 When $\mathscr{P}(G) = Q(G)$

It is interesting to find out when the LP relaxation $Q(G)$ is tight. This question is answered in Theorem 14 below.

Theorem 14 ([94, 19])**.** *The equality $\mathscr{P}(G) = Q(G)$ holds if and only if $\alpha(G) \leq 2$.*

It should be noted that the description of $Q(G)$ can involve exponentially many inequalities, even when $\alpha(G) = 2$ [94].

Proposition 7. *The following inclusions hold and are sharp.*

$$\mathscr{P}(G) \subseteq Q(G) \subseteq [0,1]^n \subseteq \alpha(G)\mathscr{P}(G).$$

Proof. The first two inclusions are trivial. Consider $x^* \in [0,1]^n$ and a facet-defining inequality $\sum_{i \in V} \pi_i x_i \leq \pi_0$ of $\mathscr{P}(G)$. Let $S = \{i \in V \mid \pi_i > 0\}$.

If $|S| = 0$, then $\pi_0 = 0$, since otherwise no feasible point could satisfy it at equality. By Lemma 3, the inequality must a nonnegativity bound $\pi_j x_j \leq 0$, in which case

$$\sum_{i \in V} \pi_i x_i^* = \pi_j x_j^* \leq 0 = \alpha(G)\pi_0.$$

Now suppose $|S| \geq 1$, so $\pi_0 > 0$. Then S must be an independent set. For each vertex $i \in S$, we have $\pi_i \leq \pi_0$ and $x_i^* \leq 1$, so

$$\sum_{i \in V} \pi_i x_i^* \leq \sum_{i \in S} \pi_i x_i^* \leq |S|\pi_0 \leq \alpha(G)\pi_0.$$

Thus $x^* \in \alpha(G)\mathscr{P}(G)$. The inclusions are sharp for the complete graph K_n, since $\alpha(K_n) = 1$. $\square$

Proposition 7 shows that the 0-1 bounds—and hence $Q(G)$—approximate $\mathscr{P}(G)$ well when the graph is very dense. This is not true for sparse graphs. Indeed, for the star graph $K_{1,n}$ and any

$\epsilon > 0$, we have

$$Q(K_{1,n}) \not\subseteq \left(\frac{1}{2} - \epsilon\right) \alpha(K_{1,n}) \mathscr{P}(K_{1,n}). \tag{4.14}$$

This is demonstrated as follows. Let v be the center vertex of $K_{1,n}$, and let the leaves be numbered $1, \ldots, n$. Consider the valid inequality (cf. Theorem 19)

$$(1-n)x_v + \sum_{i=1}^{n} x_i \leq 1.$$

and the point $y = (y_v, y_1, \ldots, y_n) = (0, \frac{1}{2}, \ldots, \frac{1}{2})$. The point y is feasible for $Q(K_{1,n})$, but it does not belong to $\left(\frac{1}{2} - \epsilon\right) \alpha(K_{1,n}) \mathscr{P}(K_{1,n})$, since

$$(1-n)y_v + \sum_{i=1}^{n} y_i = \frac{n}{2} > \left(\frac{1}{2} - \epsilon\right) n = \left(\frac{1}{2} - \epsilon\right) \alpha(K_{1,n}).$$

In this sense, $Q(G)$ provides an $O(\alpha(G))$ approximation for $\mathscr{P}(G)$ but no better.

4.3 3-Plex and 3-Defective Cliques

$G = (V, E)$ is called a s-plex if the degree of every vertex v satisfies $\deg(v) \geq |V| - s$. G is called a s-defective clique if $|E(\bar{G})| \leq s$. We consider both cases with $s = 3$.

For a 3-defective cliuqe G, if G is also a 2-defective clique, obviously $\alpha(G) \leq 2$ and in the previous section, we have described full description of $\mathscr{P}(G)$. Otherwise, there exists an independent set $\{u, v, w\} \subseteq V$. Thus $(u, v), (u, w), (v, w) \in E(\bar{G})$ and as $|E(\bar{G})| \leq 3$, $E(\bar{G}) = \{(u, v), (u, w), (v, w)\}$. At this time, we derive the following proposition. Its proof involves the following lemmata.

Lemma 7 ([94, 19])**.** *In a facet-defining inequality $\sum_{i \in V} \pi_i x_i \leq \pi_0$ of $\mathscr{P}(G)$, no pair of adjacent vertices can have positive coefficients.*

Lemma 8 ([94, 19])**.** *Suppose that $\sum_{i \in V} \pi_i x_i \leq \pi_0$ induces a facet of $\mathscr{P}(G)$. If π_u, π_v, and π_0 are its only positive coefficients, then $\pi_u = \pi_v = \pi_0$.*

Proposition 8. *Let* $G = (V, E)$ *with* $E(\bar{G}) = \{(u, v), (u, w), (v, w)\}$*, then* $\mathscr{P}(G) =$ $\{x \in [0, 1]^n | x$ *satisfies* (4.15), (4.16), (4.17), (4.18)$\}$.

$$x_u + x_v - \sum_{s \in V \setminus \{u,v,w\}} x_s \leq 1 \tag{4.15}$$

$$x_u + x_w - \sum_{s \in V \setminus \{u,v,w\}} x_s \leq 1 \tag{4.16}$$

$$x_v + x_w - \sum_{s \in V \setminus \{u,v,w\}} x_s \leq 1 \tag{4.17}$$

$$x_u + x_v + x_w - 2 \sum_{s \in V \setminus \{u,v,w\}} x_s \leq 1 \tag{4.18}$$

Proof. For any facet-defining inequality $\sum_{i \in V} \pi_i x_i \leq \pi_0$ expect for $0 - 1$ bounds, we show it is in form (4.15), (4.16), (4.17) or (4.18).

We first show $\pi_u, \pi_v, \pi_w \geq 0$. In fact, if $\pi_u < 0$, for any connected $G[D]$, let $D' = D \setminus \{u\}$, then $G[D']$ is still connected, so $\sum_{i \in V \setminus \{u\}} \pi_i x_i^D = \sum_{i \in V} \pi_i x_i^{D'} \leq \pi_0$. Thus, $\sum_{i \in V \setminus \{u\}} \pi_i x_i \leq \pi_0$ is valid and $\sum_{i \in V} \pi_i x_i \leq \pi_0$ is implied by $\sum_{i \in V \setminus \{u\}} \pi_i x_i \leq \pi_0$ and $\pi_u x_u \leq 0$. Then by Lemma 2, $\sum_{i \in V} \pi_i x_i \leq \pi_0$ is not facet-defining. This contradiction shows $\pi_u \geq 0$, and by the same reason, $\pi_v, \pi_w \geq 0$.

Next, for any $s \in V \setminus \{u, v, w\}$, let $D = \{u, v, w, s\}$, then $G[D]$ is connected and thus $\pi_u + \pi_v + \pi_w + \pi_s \leq \pi_0$. Let $k = \pi_0 - \pi_u - \pi_v - \pi_w$, so $\pi_s \leq k$. If $\pi_s < k$, obviously $\sum_{i \in V \setminus \{s\}} \pi_i x_i + k x_s \leq \pi_0$ is valid and as $\sum_{i \in V} \pi_i x_i \leq \pi_0$ is implied by $\sum_{i \in V \setminus \{s\}} \pi_i x_i + k x_s \leq \pi_0$ and $(\pi_s - k) x_s \leq 0$, by Lemma 2, we know $\sum_{i \in V} \pi_i x_i \leq \pi_0$ is not facet-defining. This contradiction shows $\pi_s = k = \pi_0 - \pi_u - \pi_v - \pi_w$.

Next we show $\pi_u, \pi_v, \pi_w = 0$ or π_0. Since $G[\{u\}]$, $G[\{v\}]$ and $G[\{w\}]$ are connected, this implies that $\pi_u, \pi_v, \pi_w \leq \pi_0$. If $\pi_u + \pi_v + \pi_w \leq \pi_0$, then inequalities $\pi_u x_u + \pi_v x_v + \pi_w x_w \leq \pi_0$ and $\sum_{i \in V \setminus \{u,v,w\}} \pi_i x_i \leq 0$ are valid (at least one of $x_i, i \in V \setminus \{u, v, w\} = 0$ as we suppose it is not a $0 - 1$ bound), so the aggregated inequality $\sum_{i \in V} \pi_i x_i \leq \pi_0$ cannot induce a facet (by Lemma 2). Thus, we will assume that $\pi_u + \pi_v + \pi_w > \pi_0$. For contradiction purposes, suppose that at least one of π_u,π_v and π_w is between 0 and π_0. Without loss of generality, suppose that $0 < \pi_u < \pi_0$. Define

$$\epsilon := \frac{1}{|V|} \min\{\pi_u + \pi_v + \pi_w - \pi_0, \pi_0 - \pi_u, \pi_u\}.$$

Note that $\pi_u + \pi_v + \pi_w - \pi_0 > 0$ and $0 < \pi_u < \pi_0$, so $\epsilon > 0$. Also, $\pi_u + \epsilon < \pi_0$, $\pi_u - |V|\epsilon > 0$ and for every $i \in S \setminus \{u, v, w\}$, we have $\pi_i + \epsilon = \pi_0 - \pi_u - \pi_v - \pi_w + \epsilon < 0$. Consider the following inequalities.

$$(\pi_u+\epsilon)x_u+\pi_v x_v+\pi_w x_w+\sum_{i\in V\setminus\{u,v,w\}}(\pi_i-\epsilon)x_i\le\pi_0 \tag{4.19}$$

$$(\pi_u-\epsilon)x_u+\pi_v x_v+\pi_w x_w+\sum_{i\in V\setminus\{u,v,w\}}(\pi_i+\epsilon)x_i\le\pi_0. \tag{4.20}$$

For any connected $G[D]$, when $u\notin D$, obviously (4.19) is valid and for (4.20), let $D'=D\cup\{u\}$, $G[D']$ is still connected, so

$$\begin{aligned}&(\pi_u-\epsilon)x_u^D+\pi_v x_v^D+\pi_w x_w^D+\sum_{i\in V\setminus\{u,v,w\}}(\pi_i+\epsilon)x_i^D\\&\le\pi_v x_v^D+\pi_w x_w^D+\sum_{i\in V\setminus\{u,v,w\}}\pi_i x_i^D+|V|\epsilon\\&\le\sum_{i\in V}\pi_i x_i^{D'}\le\pi_0.\end{aligned}$$

So (4.20) is also valid.

When $u\in D$, if neither of v,w belongs to D, obviously both (4.19) and (4.20) are valid; if at least one of v,w belongs to D, as $G[D]$ is connected, there exists $s\in D\cap(V\setminus\{u,v,w\})$. So,

$$\begin{aligned}&(\pi_u+\epsilon)x_u^D+\pi_v x_v^D+\pi_w x_w^D+\sum_{i\in V\setminus\{u,v,w\}}(\pi_i-\epsilon)x_i^D\\&\le(\pi_u+\epsilon)x_u^D+\pi_v x_v^D+\pi_w x_w^D+(\pi_s-\epsilon)x_s^D\\&\le\sum_{i\in V}\pi_i x_i^D\le\pi_0.\end{aligned}$$

Thus, (4.19) is valid and by the same reason (4.20) is also valid. $\sum_{i\in V}\pi_i x_i\le\pi_0$ is implied by (4.19) and (4.20), by Lemma 2, it is not facet-defining and this contradiction shows $\pi_u,\pi_v,\pi_w=0$ or π_0.

As $\pi_u+\pi_v+\pi_w>\pi_0$, at least two of π_u,π_v,π_w are π_0. If exactly two of π_u,π_v,π_w are π_0, as $\pi_s=\pi_0-\pi_u-\pi_v-\pi_w,\forall s\in V\setminus\{u,v,w\}$, the inequality is either (4.15), (4.16) or (4.17). If $\pi_u,\pi_v,\pi_w=\pi_0$, still as $\pi_s=\pi_0-\pi_u-\pi_v-\pi_w,\forall s\in V\setminus\{u,v,w\}$, the inequality is (4.18). □

For a 3-plex G, if G is also a 2-plex, still $\alpha(G)\le 2$ and in the previous section, we have derived full description of $\mathscr{P}(G)$. Otherwise, we have the following proposition.

Proposition 9. *Let $G = (V, E)$ be a 3-plex and $|V| = n$, then*

$$\mathscr{P}(G) = \{x \in [0,1]^n \ \mid x \text{ satisfies all } a, b\text{-separator inequalities and } (4.21)\}.$$

$$x_u + x_v + x_w - 2 \sum_{s \in V \setminus \{u,v,w\}} x_s \leq 1 \ \forall \text{ independent set } \{u, v, w\}. \tag{4.21}$$

Proof. For any facet-defining inequality $\sum_{i \in V} \pi_i x_i \leq \pi_0$, let $S = \{k | \pi_k > 0\}$ and $N = |S|$. If $N \geq 4$, by Lemma 7, for any $i, j \in S, (i, j) \notin E$, so for any $i \in S$, $\delta(i) \leq |V| - 4$, a contraction. So N can be $0, 1, 2, 3$.

When $N = 0$ or 1, we can show that the facet-defining inequalities are upper or lower bound.

When $N = 2$, suppose $\pi_a, \pi_b > 0$ and $\pi_i \leq 0, \forall i \neq a, b$, then by Lemma 7, a and b are not adjacent and as $\pi_0 \geq \pi_a > 0$, by Lemma 8, $\pi_a = \pi_b = \pi_0$. Because G is a 3-plex, $\deg(a), \deg(b) \geq n-3$. Note a and b are not adjacent, $|N(a) \cap N(b)| \geq n - 4$ and $|N(a) \setminus N(b)| \leq 1$, $|N(b) \setminus N(a)| \leq 1$. For any $u \in N(a) \cap N(b)$, because $G[\{a, b, u\}]$ is connected, $\pi_u \leq \pi_0 - \pi_a - \pi_b = -\pi_0$; if $\pi_u < -\pi_0$, obviously $\sum_{i \in V \setminus \{u\}} \pi_i x_i - \pi_0 x_u \leq \pi_0$ is valid and as $\sum_{i \in V} \pi_i x_i \leq \pi_0$ is implied by $\sum_{i \in V \setminus \{u\}} \pi_i x_i - \pi_0 x_u \leq \pi_0$ and $(\pi_u - \pi_0) x_u \leq 0$, by Lemma 2, we know $\sum_{i \in V} \pi_i x_i \leq \pi_0$ is not facet-defining. This contradiction shows $\pi_u = -\pi_0$.

Further, if there exists $v \in N(a) \setminus N(b)$, we have two cases.

1. $N(v) \setminus \{a\} \subseteq N(a) \cap N(b)$, then we show $\pi_v = 0$. In fact, if $\pi_v < 0$, note for any connected $G[D]$ including a and b, let $D' = D \setminus \{v\}$, $G[D']$ is still connected, so $\sum_{i \in V \setminus \{u\}} \pi_i x_i^D = \sum_{i \in V} \pi_i x_i^{D'} \leq \pi_0$; for any connected $G[D]$ not including both a and b, $\sum_{i \in V \setminus \{v\}} \pi_i x_i^D \leq \pi_0$, so $\sum_{i \in V \setminus \{v\}} \pi_i x_i \leq \pi_0$ is valid and $\sum_{i \in V} \pi_i x_i \leq \pi_0$ is implied by $\sum_{i \in V \setminus \{v\}} \pi_i x_i \leq \pi_0$ and $\pi_v x_v \leq 0$. Then by Lemma 2, $\sum_{i \in V} \pi_i x_i \leq \pi_0$ is not facet-defining. This contradiction shows $\pi_v = 0$ (as we suppose $\pi_v \leq 0$).

2. $N(v) \setminus \{a\} \nsubseteq N(a) \cap N(b)$, then there exists $w \in N(b) \cap N(v) \setminus N(a)$. As $|N(a) \setminus N(b)| \leq 1$ and $|N(b) \setminus N(a)| \leq 1$, $u \in N(a) \cap N(b), \forall u \in V \setminus \{a, b, v, w\}$. By the same arguments as

above, we have $\pi_v + \pi_w = -\pi_0$. If $-\pi_0 < \pi_v, \pi_w < 0$, obviously both

$$\pi_0 x_a + \pi_0 x_b - \pi_0 x_v + \sum_{i \in V \setminus \{a,b,v,w\}} \pi_i x_i \leq \pi_0$$
$$\pi_0 x_a + \pi_0 x_b - \pi_0 x_w + \sum_{i \in V \setminus \{a,b,v,w\}} \pi_i x_i \leq \pi_0$$

are valid and $\sum_{i \in V} \pi_i x_i \leq \pi_0$ is implied by these two inequalities. So by Lemma 2, $\sum_{i \in V} \pi_i x_i \leq \pi_0$ is not facet-defining. This contradiction shows $\pi_v = -\pi_0$, $\pi_w = 0$ or $\pi_w = -\pi_0$, $\pi_v = 0$.

So, in general the facet-defining inequality can be simplified as $x_a + x_b - \sum_{v_i \in C} x_i \leq 1$, where $C = \{u | \pi_u = -\pi_0\}$. By Theorem 9, C is a minimal $(a,b)-$separator. So the inequality is an a,b-separator inequality.

When $N = 3$, suppose $\pi_u, \pi_v, \pi_w > 0$ and $\pi_i \leq 0, \forall i \neq u, v, w$. Note $|N(u)| \geq n - 3$ but $v, w \notin N(u)$, so $s \in N(u), \forall s \in V \setminus \{u, v, w\}$. For the same reason, $s \in N(u) \cap N(v) \cap N(w), \forall s \in V \setminus \{u, v, w\}$. Thus, the arguments in Proposition 8 still holds and the facet-defining inequalities have the form (4.21).

So, the proposition is true in general. $\square$

4.4 The Case of Forests

In this section, we focus on $\mathscr{P}(G)$ in the case that G is a forest. The objective is to show that every facet-defining inequality of $\mathscr{P}(G)$ can be generated via lifting from the seed inequality $x_i \leq 1$ (with every other variable initially fixed to zero). This does not hold for arbitrary graphs. Moreover, for any lifting order, the *entire* facet-defining inequality can be generated in time $O(n)$. Along the way, we show several interesting structural properties about $\mathscr{P}(G)$. We conclude by providing closed-form descriptions of $\mathscr{P}(G)$ for path and star graphs.

Lemma 9. *Consider a tree $G = (V, E)$ with $|V| \geq 2$ and a facet-defining inequality $\sum_{i \in V} \pi_i x_i \leq \pi_0$ of $\mathscr{P}(G)$ with $\pi_0 > 0$. If v is a leaf, then $\pi_v \geq 0$.*

Proof. For contradiction purposes, suppose that v is a leaf with $\pi_v < 0$. Note that $\pi_v x_v \leq 0$ is valid. We argue that $0 x_v + \sum_{i \in V \setminus \{v\}} \pi_i x_i \leq \pi_0$ is also valid. Suppose that $D \subseteq V$ induces a connected subgraph. Then, since v is a leaf, $G[D']$ is also connected, where $D' = D \setminus \{v\}$. Thus,

$$0x_v^D + \sum_{i \in V \setminus \{v\}} x_i^D = \sum_{i \in V} \pi_i x_i^{D'} \leq \pi_0,$$

so the new inequality is valid. Further, it is not the $0x \leq 0$ inequality. The inequality $\sum_{i \in V} \pi_i x_i \leq \pi_0$ is distinct from the new inequality and from $\pi_v x_v \leq 0$, yet it is implied by them. Thus, Lemma 2 shows that $\sum_{i \in V} \pi_i x_i \leq \pi_0$ cannot induce a facet, a contradiction. □

Lemma 10. *Consider a tree $G = (V, E)$ with $|V| \geq 2$, a leaf v of G, its stem s, and a facet-defining inequality $\sum_{i \in V} \pi_i x_i \leq \pi_0$ of $\mathscr{P}(G)$ with $\pi_0 > 0$. Then for any $a \in [0, \pi_0]$, the following inequality is valid.*

$$a x_v + (\pi_v + \pi_s - a) x_s + \sum_{i \in V \setminus \{v,s\}} \pi_i x_i \leq \pi_0. \tag{4.22}$$

Moreover, $(\pi_v + \pi_s) x_s + \sum_{i \in V \setminus \{v,s\}} \pi_i x_i \leq \pi_0$ induces a facet of $\mathscr{P}(G - v)$.

Proof. First we show that inequality (4.22) is valid. Consider a subtree $G[D]$ of G. We consider two cases.

In the first case, suppose that $s \notin D$. Then $D = \{v\}$ or $v \notin D$. If $D = \{v\}$, then

$$a x_v^D + (\pi_v + \pi_s - a) x_s^D + \sum_{i \in V \setminus \{v,s\}} \pi_i x_i^D = a \leq \pi_0,$$

and if $v \notin D$, then

$$a x_v^D + (\pi_v + \pi_s - a) x_s^D + \sum_{i \in V \setminus \{v,s\}} \pi_i x_i^D = \sum_{i \in V} \pi_i x_i^D \leq \pi_0.$$

Thus, if $s \notin D$, then the inequality is valid.

In the second case, suppose that $s \in D$. Then $G[D']$ is connected, where $D' = D \cup \{v\}$, so

$$\begin{aligned}
&ax_v^D + (\pi_v + \pi_s - a)x_s^D + \sum_{i \in V \setminus \{v,s\}} \pi_i x_i^D \\
&\leq ax_v^{D'} + (\pi_v + \pi_s - a)x_s^{D'} + \sum_{i \in V \setminus \{v,s\}} \pi_i x_i^{D'} \\
&= \sum_{i \in V} \pi_i x_i^{D'} \leq \pi_0.
\end{aligned}$$

Thus, inequality (4.22) is valid in all cases.

By setting $a = 0$, we see that the inequality

$$(\pi_v + \pi_s)x_s + \sum_{i \in V \setminus \{v,s\}} \pi_i x_i \leq \pi_0 \tag{4.23}$$

is valid for $\mathscr{P}(G)$. We argue that inequality (4.23) induces a facet of $\mathscr{P}(G - v)$. It is valid by Lemma 1. Because $\mathscr{P}(G)$ is full-dimensional and by assumption that $\sum_{i \in V} \pi_i x_i \leq \pi_0$ induces a facet of $\mathscr{P}(G)$, there exist n affinely independent points $x^{D_1}, \ldots, x^{D_n}$ satisfying $\sum_{i \in V} \pi_i x_i^{D_j} = \pi_0$. For each x^{D_j}, delete the element in position v to get an $n-1$ dimensional vector y^j.

Now we argue that the y^j vectors can be used to show that the face where inequality (4.23) holds at equality has dimension $n - 2$. By Lemma 9, $\pi_v \geq 0$. We consider two cases. In the first case, $\pi_v = 0$. Then for each y^j,

$$(\pi_v + \pi_s)y_s^j + \sum_{i \in V \setminus \{v,s\}} \pi_i y_i^j = \sum_{i \in V} \pi_i x_i^{D_j} = \pi_0.$$

Because $\mathrm{rank}(x^{D_2} - x^{D_1}, \ldots, x^{D_n} - x^{D_1}) = n - 1$, we have $\mathrm{rank}(y^2 - y^1, \ldots, y^n - y^1) \geq n - 2$, so there are $n-1$ affinely independent vectors among $y^1, \ldots, y^n$ and each satisfies inequality (4.23) at equality, so inequality (4.23) induces a facet of $\mathscr{P}(G - v)$ as $\dim(\mathscr{P}(G - v)) = n - 1$.

Now suppose $\pi_v > 0$. Then for any x^{D_j} consider the following cases.

1. $x_v^{D_j} = x_s^{D_j}$. In this case,

$$\begin{aligned}
(\pi_v + \pi_s)y_s^j + \sum_{i \in V \setminus \{v,s\}} \pi_i y_i^j &= (\pi_v + \pi_s)x_s^{D_j} + \sum_{i \in V \setminus \{v,s\}} \pi_i x_i^{D_j} \\
&= \sum_{i \in V} \pi_i x_i^{D_j} = \pi_0.
\end{aligned}$$

2. $x_v^{D_j} = 1$ and $x_s^{D_j} = 0$. Then $D_j = \{v\}$.

3. $x_v^{D_j} = 0$ and $x_s^{D_j} = 1$, let $D'_j = D_j \cup \{v\}$, then D'_j is still connected and $\sum_{i \in V} \pi_i x_i^{D'_j} = \sum_{i \in V} \pi_i x_i^{D_j} + \pi_v > \pi_0$, so this case cannot happen.

So if x^{D_j} is not the zero vector with a one in position v, then $x_v^{D_j} = x_s^{D_j}$. Thus there must be $n-1$ affinely independent vectors $x^{D_1}, \ldots, x^{D_{n-1}}$ for which $x_v^{D_j} = x_s^{D_j}$. As $x_v^{D_j} = x_s^{D_j}$ for $j = 1, \ldots, n-1$ and by definition of y^j, we have

$$\operatorname{rank}(y^2 - y^1, \ldots, y^{n-1} - y^1) = \operatorname{rank}(x^{D_2} - x^{D_1}, \ldots, x^{D_{n-1}} - x^{D_1}) = n - 2,$$

and, as shown above, each y^j satisfies inequality (4.23) at equality. Thus, inequality (4.23) induces a facet of $\mathscr{P}(G - v)$ when $\pi_v > 0$. This concludes the proof. □

Lemma 11. *Consider a tree $G = (V, E)$ and a facet-defining inequality $\sum_{i \in V} \pi_i x_i \leq \pi_0$ of $\mathscr{P}(G)$ with $\pi_0 > 0$. Then, for each $i \in V$, π_0 divides π_i.*

Proof. The proof is by induction on $|V|$. In the base case, where $|V| = 1$, the inequality must be a multiple of $x_1 \leq 1$, so the statement is true. Now suppose the statement holds when $|V| = k - 1$ and consider a tree with $|V| = k \geq 2$. Since G is a tree with $|V| \geq 2$, it has a leaf v, and consider its stem s. By Lemma 9, $\pi_v \geq 0$. Since $G[\{v\}]$ is connected, $\pi_v \leq \pi_0$.

Suppose $0 < \pi_v < \pi_0$. Then by Lemma 10, both of the following inequalities are valid for $\mathscr{P}(G)$.

$$(\pi_v + \pi_s) x_s + \sum_{i \in V \setminus \{v,s\}} \pi_i x_i \leq \pi_0$$

$$\pi_0 x_v + (\pi_v + \pi_s - \pi_0) x_s + \sum_{i \in V \setminus \{v,s\}} \pi_i x_i \leq \pi_0.$$

By multiplying the first inequality by $1 - \frac{\pi_v}{\pi_0}$ and the second by $\frac{\pi_v}{\pi_0}$ and by Lemma 2, $\sum_{i \in V} \pi_i x_i \leq \pi_0$ cannot be facet-defining. The contradiction shows $\pi_v = 0$ or $\pi_v = \pi_0$. In both cases, π_0 divides π_v.

Now we must show that other coefficients are also divisible by π_0. By Lemma 10, $(\pi_v + \pi_s) x_s + \sum_{i \in V \setminus \{v,s\}} \pi_i x_i \leq \pi_0$ induces a facet of $\mathscr{P}(G - v)$. Because $|V(G - v)| = k - 1$ and by the induction assumption, π_0 divides π_i for each $i \in V \setminus \{v, s\}$. Also π_0 divides $\pi_v + \pi_s$, implying that it divides π_s. Thus, every coefficient of the inequality $\sum_{i \in V} \pi_i x_i \leq \pi_0$ is divisible by π_0 when $|V| = k$, and the statement is true in general. □

Remark 1. *By Lemma 11, we can suppose that if G is a tree and $\pi_0 > 0$, then $\pi_0 = 1$ and each π_i is an integer.*

Lemma 12. *Consider a tree $G = (V, E)$ and a valid inequality $\sum_{i \in V} \pi_i x_i \leq 1$ for $\mathscr{P}(G)$. If $\sum_{i \in V} \pi_i = 1$ and every π_i is an integer, then there is a vertex $v \in V$ such that $\pi_v = 1 - |N_G(v)|$.*

Proof. If $|V| = 1$, then the statement clearly holds, so consider $|V| \geq 2$. First we show that $\pi_i \geq 1 - |N_G(i)|$ for every vertex $i \in V$. Suppose there is a vertex $v \in V$ with $\pi_v < 1 - |N_G(v)|$. We can partition $V \setminus \{v\}$ into $|N_G(v)|$ sets—specifically the vertex sets of the components $\{G_j = (V_j, E_j)\}_j$ of $G - v$. Since $\sum_{i \in V} \pi_i x_i \leq 1$ is valid, and since each G_j is connected, $\sum_{u \in V_j} \pi_u = \sum_{i \in V} \pi_i x_i^{V_j} \leq 1$. Then,

$$\sum_{i \in V} \pi_i = \pi_v + \sum_j \sum_{u \in V_j} \pi_u \leq \pi_v + |N_G(v)| < 1,$$

which contradicts that $\sum_{i \in V} \pi_i = 1$. So $\pi_i \geq 1 - |N_G(v_i)|$ for every $i \in V$.

Suppose there is no vertex v satisfying $\pi_v = 1 - |N_G(v)|$. Then, $\pi_i \geq 2 - |N_G(i)|$ for every $i \in V$. Let L denote the set of leaves of G. For any leaf $l \in L$, we have $\pi_l = 1$. This follows because $\pi_l \geq 2 - |N_G(l)| = 1$, and $\pi_l \leq 1$ since $\{l\}$ is connected. Because G is a tree,

$$\sum_{i \in V \setminus L} |N_G(i)| = 2|V \setminus L| + |L| - 2.$$

Thus, we have

$$\begin{aligned} 1 = \sum_{i \in V} \pi_i &= \sum_{i \in V \setminus L} \pi_i + |L| \\ &\geq 2|V \setminus L| - \sum_{i \in V \setminus L} |N_G(i)| + |L| \\ &= 2|V \setminus L| - (2|V \setminus L| + |L| - 2) + |L| = 2. \end{aligned}$$

This contradiction shows there must exist $v \in V$ such that $\pi_v = 1 - |N_G(v)|$. □

Lemma 13. *Consider a tree $G = (V, E)$ with $|V| \geq 2$ and a facet-defining inequality $\sum_{i \in V} \pi_i x_i \leq 1$ of $\mathscr{P}(G)$. If $\sum_{i \in V} \pi_i = 1$ and $\pi_v = 1 - |N_G(v)|$, then each component $G' = (V', E')$ of $G - v$ satisfies $\sum_{i \in V'} \pi_i = 1$.*

Proof. As in the proof of Lemma 12, we can partition $V \setminus \{v\}$ into $|N_G(v)|$ sets—the vertex sets of the components $\{G_j = (V_j, E_j)\}_j$ of $G - v$. Then,

$$1 = \sum_{i \in V} \pi_i = \pi_v + \sum_j \sum_{i \in V_j} \pi_i \leq \pi_v + |N_G(v)| = 1,$$

where the inequality holds since $G[V_j]$ is connected and since $\sum_{i \in V_j} \pi_i x_i \leq 1$ is valid (by Lemma 1). So, for each V_j, we must have $\sum_{i \in V_j} \pi_i = 1$. □

Lemma 14. *Consider a forest $G = (V, E)$, a vertex $v \in V$, and a facet-defining inequality $\sum_{i \in V \setminus \{v\}} \pi_i x_i \leq 1$ of $\mathscr{P}(G - v)$. Then, when lifting in v, the resulting facet-defining inequality $\sum_{i \in V} \pi_i x_i \leq 1$ of $\mathscr{P}(G)$ has $\pi_v = 1 - |N_G(v)|$.*

Proof. When solving the lifting problem, $\zeta \leq |N_G(v)|$ by Lemma 4. The lower bound $\zeta \geq |N_G(v)|$ holds by choosing v and all components of $G - v$ that v neighbors in G (each component has weight 1 by Lemma 13). Thus, by Corollary 3, when $\pi_v = 1 - \zeta = 1 - |N_G(v)|$, the inequality induces a facet. □

Lemma 15. *For a tree $G = (V, E)$, the inequality $\sum_{i \in V} \pi_i x_i \leq 1$ induces a facet of $\mathscr{P}(G)$ if and only if*

1. *$\sum_{i \in V} \pi_i x_i \leq 1$ is valid for $\mathscr{P}(G)$,*
2. *for each $i \in V$, π_i is an integer, and*
3. *$\sum_{i \in V} \pi_i = 1$.*

Proof. ($\Longleftarrow$) Suppose $\sum_{i \in V} \pi_i x_i \leq 1$ is valid, each π_i is an integer, and $\sum_{i \in V} \pi_i = 1$. We show the inequality is facet-defining by induction on $|V|$. When $|V| = 1$, the only inequality satisfying the conditions is $x_i \leq 1$ and it is facet-defining by Proposition 4, so the statement is true.

Now suppose the statement is true when $|V| < k$, and consider $|V| = k$. By Lemma 12, there is a vertex $v \in V$ such that $\pi_v = 1 - |N_G(v)|$. By Lemma 13, each component $G_j = (V_j, E_j)$ satisfies $\sum_{i \in V_j} \pi_i = 1$. Also, $\sum_{i \in V_j} \pi_i x_i \leq 1$ is valid for $G[V_j]$ by Lemma 1 and $|V_j| < k$. So, by the induction assumption, for each V_j, the inequality $\sum_{i \in V_j} \pi_i x_i \leq 1$ induces a facet of $\mathscr{P}(G[V_j])$. Theorem 8 then implies that $\sum_j \sum_{i \in V_j} \pi_i x_i \leq 1$ induces a facet of $\mathscr{P}(G - v)$. Lemma 14 shows that for $\pi_v = 1 - |N_G(v)|$, the inequality $\sum_{i \in V} \pi_i x_i \leq 1$ induces a facet of $\mathscr{P}(G)$. So the statement is true when $|V| = k$, and it is true in general.

($\Longrightarrow$) Suppose $\sum_{i \in V} \pi_i x_i \leq 1$ induces a facet of $\mathscr{P}(G)$. Then it is valid and by Lemma 11, each π_i is an integer. We use induction to show that $\sum_{i \in V} \pi_i = 1$.

When $|V| = 1$, the only facet-defining inequality is $x_i \leq 1$, so the statement is true. Now suppose the statement is true when $|V| < k$, and consider $|V| = k$. Let v be a leaf and s be its stem. By Lemma 10, the inequality

$$(\pi_v + \pi_s)x_s + \sum_{i \in V \setminus \{v,s\}} \pi_i x_i \leq 1$$

induces a facet of $\mathscr{P}(G - v)$. Then, by the induction assumption,

$$\sum_{i \in V} \pi_i = \sum_{i \in V \setminus \{v,s\}} \pi_i + (\pi_v + \pi_s) = 1.$$

So, the statement is true when $|V| = k$, and it is true in general. $\square$

Theorem 15. *For a forest $G = (V, E)$, the inequality $\sum_{i \in V} \pi_i x_i \leq 1$ induces a facet of $\mathscr{P}(G)$ if and only if each component $G' = (V', E')$ of G satisfies*

1. *$\sum_{i \in V'} \pi_i x_i \leq 1$ is valid for $\mathscr{P}(G')$,*

2. *π_i is an integer for each $i \in V'$, and*

3. *$\sum_{i \in V'} \pi_i = 1$.*

Proof. Directly from Theorem 8 and Lemma 15. $\square$

Lemma 16. *Consider a tree $G = (V, E)$ with $|V| \geq 2$ and a facet-defining inequality $\sum_{i \in V} \pi_i x_i \leq 1$ of $\mathscr{P}(G)$. If there is a vertex $v \in V$ with $\pi_v = 1 - |N_G(v)|$, then $\sum_{i \in V \setminus \{v\}} \pi_i x_i \leq 1$ induces a facet of $\mathscr{P}(G - v)$.*

Proof. By Lemma 15, $\sum_{i \in V} \pi_i = 1$ and each π_i is an integer. So, by Lemma 12, there exists $v \in V$ such that $\pi_v = 1 - |N_G(v)|$. Consider the components $\{G_j = (V_j, E_j)\}_j$ of $G - v$. By Lemma 13, each V_j satisfies $\sum_{i \in V_j} \pi_i = 1$. Also, for each V_j, the inequality $\sum_{i \in V_j} \pi_i x_i \leq 1$ is valid for $G[V_j]$ by Lemma 1 and each π_i is an integer, so by Theorem 15, $\sum_{i \in V_j} \pi_i x_i \leq 1$ induces a facet of $\mathscr{P}(G[V_j])$. Thus, by Theorem 8, $\sum_{i \in V_j} \pi_i x_i \leq 1$ induces a facet of $\mathscr{P}(G - v)$. $\square$

Now we can state our main theorem.

Theorem 16. *For a forest $G = (V, E)$, an inequality $\sum_{i \in V} \pi_i x_i \leq \pi_0$ with $\pi_0 > 0$ induces a facet of $\mathscr{P}(G)$ if and only if it can be obtained via some lifting order from an $x_i \leq 1$ seed inequality for some $i \in V$.*

Proof. ($\Longleftarrow$) By repeatedly applying Corollary 3, i.e., sequential lifting.

($\Longrightarrow$) By Lemma 11, we can suppose that $\pi_0 = 1$ and each π_i is an integer. The proof is by induction on $|V|$. In the base case, where $|V| = 1$, the statement holds since the facet-defining inequality must be $x_i \leq 1$ for some $i \in V$.

So suppose the statement holds for $|V| < k$, and consider $|V| = k$. Let $G' = (V', E')$ be a component of G. By Theorem 15, $\sum_{i \in V'} \pi_i = 1$. Then, since $\sum_{i \in V'} \pi_i x_i \leq 1$ is valid for $\mathscr{P}(G')$ by Lemma 1, Lemma 12 shows that there exists $v \in V'$ such that $\pi_v = 1 - |N_{G'}(v)|$. By Lemma 16, the inequality $\sum_{i \in V' \setminus \{v\}} \pi_i x_i \leq 1$ induces a facet of $\mathscr{P}(G' - v)$. Then, by Theorem 8, the inequality

$$\sum_{i \in V \setminus \{v\}} \pi_i x_i \leq 1 \tag{4.24}$$

induces a facet of $G - v$. Since $G - v$ is a forest, and by the induction assumption, inequality (4.24) can be obtained via lifting from an $x_i \leq 1$ seed inequality. Thus, by Lemma 14, lifting in v by setting $\pi_v = 1 - |N_G(v)|$ results the facet-defining inequality $\sum_{i \in V} \pi_i x_i \leq 1$. So the statement is true when $|V| = k$, and it is true in general. ☐

Theorem 17. *Given a forest $G = (V, E)$ and a vertex order $(i_1, \ldots, i_n)$, the entire sequentially-lifted facet-defining inequality for $\mathscr{P}(G)$ can be generated in time $O(n)$ from the seed inequality $x_{i_1} \leq 1$.*

Proof. Here is the algorithm.

1. $S \leftarrow \emptyset$;
2. **for** $j = 1, \ldots, n$ **do**
   - $\pi_{i_j} \leftarrow 1 - |N_G(i_j) \cap S|$;
   - $S \leftarrow S \cup \{i_j\}$;
3. **return** inequality $\sum_{i \in V} \pi_i x_i \leq 1$;

It can be shown by induction on k and by Lemma 14 that, in iteration k of the loop, the inequality $\sum_{j=1}^{k} \pi_{i_j} x_{i_j} \leq 1$ induces a facet of $\mathscr{P}(G[\cup_{j=1}^{k}\{i_j\}])$. So the final inequality induces a facet of $\mathscr{P}(G)$.

To achieve a runtime of $O(n)$, we can represent G in adjacency list format and S as a boolean n-vector. In iteration j of the loop, we compute π_{i_j} by counting the number of i_j's neighbors that belong to S. Since G is a forest, the total number of neighbors to check over all iterations is less than $2n$. □

Theorem 18 ($\mathscr{P}(G)$ for path graphs)**.** *Consider the n-vertex path graph*

$$P_n = ([n], \{\{1,2\}, \ldots, \{n-1, n\}\}). \tag{4.25}$$

For any odd-length subsequence $(i_1, \ldots, i_q)$ of $(1, \ldots, n)$, the following path *inequality induces a facet of $\mathscr{P}(P_n)$.*

$$\sum_{\substack{j \in [q] \\ j \text{ odd}}} x_{i_j} - \sum_{\substack{j \in [q] \\ j \text{ even}}} x_{i_j} \leq 1. \tag{4.26}$$

Moreover, the nonnegativity bounds and path inequalities (4.26) fully describe $\mathscr{P}(P_n)$. Consequently, the number of facets of $\mathscr{P}(P_n)$ is

$$n + \sum_{\substack{k \in [n] \\ k \text{ odd}}} \binom{n}{k}.$$

Proof. Each coefficient of a path inequality (4.26) is integer, and their sum is 1. So, by Theorem 15, to show that inequality (4.26) induces a facet, we only need to show that it is valid. This is done by induction on q (incrementing by 2 each time). When $q = 1$ or $q = 3$ it is obviously valid. Suppose it holds for odd $q < k$, and consider odd $q = k \geq 5$. By the induction assumption, the following path inequalities are valid.

$$\sum_{\substack{j\in[q-2]\\ j\text{ odd}}} x_{i_j} - \sum_{\substack{j\in[q-2]\\ j\text{ even}}} x_{i_j} \le 1$$

$$\sum_{\substack{j\in[q]\setminus\{1,2\}\\ j\text{ odd}}} x_{i_j} - \sum_{\substack{j\in[q]\setminus\{1,2\}\\ j\text{ even}}} x_{i_j} \le 1$$

$$x_{i_1} - x_{i_2} + x_{i_q} \le 1.$$

Then, the Chvátal-Gomory procedure, c.f. [72], applied to these inequalities with each weight equal to 1/2 shows that the path inequality (4.26) is valid for $\mathscr{P}(P_n)$.

Now, we show that if a facet-defining inequality $\sum_{i=1}^n \pi_i x_i \le \pi_0$ of $\mathscr{P}(P_n)$ is not a nonnegativity bound, then it is a path inequality (4.26). By Lemmata 3 and 11, we can suppose that $\pi_0 = 1$ and each π_i is integer. Further, by Theorem 15, $\sum_{i=1}^n \pi_i = 1$. Let $S = \{i \in [n] \mid \pi_i > 0\}$ denote the set of vertices with positive coefficients. Since the coefficients are integer and bounded by $\pi_0 = 1$, this means each $i \in S$ has $\pi_i = 1$. For any pair of vertices $u, v \in S$ with $u < v$ such that no $k \in S$ satisfies $u < k < v$, the path V_{uv} connecting them is feasible, so $\sum_{i=1}^n \pi_i x_i^{V_{uv}} \le 1$. This implies a vertex k between them with $\pi_k < 0$, and, in fact, $\pi_k \le -1$ by integrality of π. Further, k is the unique vertex between u and v with $\pi_k \le -1$, and the equality $\pi_k = -1$ must hold. If otherwise, then we would have $\sum_{i=1}^n \pi_i < 1$, a contradiction. Thus, $\sum_{i=1}^n \pi_i x_i \le 1$ must be a path inequality (4.26).

The number of nonnegativity bounds is n, and the number of path inequalities is $\sum_{\substack{k\in[n]\\ k\text{ odd}}} \binom{n}{k}$, so the total number of facets of $\mathscr{P}(P_n)$ is as stated. □

Theorem 19 ($\mathscr{P}(G)$ for star graphs)**.** *Consider the $(n+1)$-vertex star graph $K_{1,n}$ with vertex set $V = \{v\} \cup [n]$, where v is the center vertex. For any $S \subseteq [n]$, the following* star *inequality induces a facet of $\mathscr{P}(K_{1,n})$.*

$$(1 - |S|)x_v + \sum_{i\in S} x_i \le 1. \tag{4.27}$$

Moreover, the nonnegativity bounds and star inequalities fully describe $\mathscr{P}(K_{1,n})$. Consequently, the number of facets of $\mathscr{P}(K_{1,n})$ is $2^n + n + 1$.

Proof. By Theorem 15, to prove that inequality (4.27) induces a facet, it is enough to show that it is valid. Consider a vertex subset D that induces a connected subgraph. If $v \notin D$, then D must be empty or a single vertex, so the inequality holds trivially. Otherwise, $v \in D$, and the inequality reduces to $\sum_{i \in S} x_i \leq |S|$, which is also trivially satisfied.

Now we show that if a facet-defining inequality $\sum_{i \in V} \pi_i x_i \leq \pi_0$ is not a nonnegativity bound, then it is a star inequality (4.27). By Lemmata 3 and 11, we can suppose that $\pi_0 = 1$ and each π_i is integer. Further, by Theorem 15, $\sum_{i \in V} \pi_i = 1$.

Let $S = \{i \in [n] \mid \pi_i > 0\}$. Then each $i \in S$ has $\pi_i = 1$. The subgraph induced by $S' = S \cup \{v\}$ is connected, and $\pi_v + |S| = \sum_{i \in V} \pi_i x_i^{S'} \leq 1$, so $\pi_v \leq 1 - |S|$. Since $\pi_i \leq 0$ for each $i \in [n] \setminus S$, the only way to achieve $\sum_{i \in V} \pi_i = 1$ is to have $\pi_v = 1 - |S|$ and $\pi_i = 0$ for each $i \in [n] \setminus S$. Thus, the inequality is a star inequality (4.27), as desired.

The number of star inequalities is $\sum_{k=0}^{n} \binom{n}{k} = 2^n$, and the number of nonnegativity bounds is $n + 1$, so the total number of facets of $\mathscr{P}(K_{1,n})$ is $2^n + n + 1$. $\square$

5. 2-CLIQUES IN UNIT DISK GRAPHS

We change our focus to clique relaxations in unit disk graphs in this chapter. As discussed in the introduction, clique and clique relaxations have direct correspondence to connectivity and the class of unit disk graphs is an important tool to model geometric networks. In this dissertation, we discuss a clique relaxation problem called the *maximum 2-clique* problem in unit disk graphs. We first introduce our motivation of this problem in Section 5.1 and definitions and notations in addition to background chapter in Section 5.2. Then in Section 5.3 we present our main observation to solve the 2-clique problem, which states that any 2-clique in a unit disk graph is dominated by no more than 4 vertices. We also mention it is not the case for general graphs in section 5.4. Finally, in Section 5.5 we discuss how to solve the maximum 2-clique problem effectively in unit disk graphs using our observation, ultimately establishing a $\frac{1}{2}$-approximation ratio for our polytime algorithm, as well as how the proposed method performs on random unit disk graphs, showing both theoretical and computational results.

5.1 Motivations

A *unit disk graph* (UDG), which can be defined as the intersection graph of closed disks of equal (e.g., unit) diameter, provide a convenient modeling tool for wireless networks, where the ability of two wireless nodes to communicate depends on whether they are within the unit Euclidean distance away from each other. While many of the classical optimization problems on graphs, such as the maximum independent set, minimum vertex cover, graph coloring, minimum dominating set, and minimum connected dominating set problems, remain NP-hard when restricted to UDGs [26], there are some notable exceptions. In particular, the maximum clique problem, which is NP-hard in general, is polynomially solvable in UDGs. This can be shown as follows [26]. Let $r \leq 1$ be the largest Euclidean distance between a pair of nodes, a and b, of disks belonging to a maximum clique $\mathcal{C}$ of a given UDG. Then all nodes of $\mathcal{C}$ must belong to the area of overlap (referred to as a *lens*) of a pair of disks of radius r centered at a and b, respectively (see Fig. 5.1). Observe that the line between a and b bisects the lens into two half-lenses, such that any two points from the same half-lens are distance at most r from each other. Hence, the set of nodes of the UDG belonging to the same half-lens forms a clique, implying that the nodes located in the lens induce a co-bipartite graph (i.e., the compliment of a bipartite graph). Since the maximum clique problem

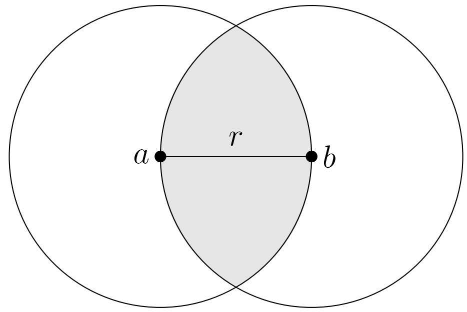


Figure 5.1: The lens formed by two disks with distance $r \leq 1$ between their centers a and b.

in a co-bipartite graph is equivalent to the maximum independent set problem in a bipartite graph, it can be solved in $O(n^{2.5})$ time, where n is the number of nodes [51]. Thus, by computing a maximum clique for every such "lens subgraph", we can solve the maximum clique problem in the original UDG in $O(n^{4.5})$ time [26]. The reader is referred to [17, 84, 6] for more information on algorithms for the maximum clique problem in UDGs.

The present work is motivated by the practical importance of UDGs and increasing interest in *clique relaxation* models, which relax various elementary graph-theoretic properties implicitly enforced by the definition of clique in order to obtain structures that are less restrictive than cliques, but are still sufficiently cohesive for a particular application of interest [82]. Several such models have been studied in the literature from the optimization perspective, including the edge density-based relaxation called γ-quasi-clique [79]; the degree-based clique relaxations called s-defective clique and s-plex [96, 7]; and the distance-based relaxations known as s-clique and s-clubs [8, 92]. To the best of our knowledge, none of the corresponding optimization problems have been studied in UDGs. Since small distance between the nodes is one of the key requirements in designing routing protocols in wireless communication networks [62], the distance-based clique relaxations restricted to UDGs deserve a special attention. Hence, this dissertation focusses on the maximum 2-clique and 2-club problems in UDGs.

5.2 Definitions and Notations

This section introduces definitions, notation, and some known facts used in the dissertation. To avoid ambiguity in terminology, we will use the term *node* to describe a vertex of a graph, while reserving the term *vertex* for points in $\mathbb{R}^2$ representing certain geometric objects such as angles, triangles, and lenses. In addition, we will reserve the term *distance* for graph-theoretic distances; whenever we will refer to Euclidean distances, we will explicitly state so. We will denote the given UDG by $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is its set of nodes and $\mathcal{E}$ is its set of edges. We will use the *containment model* of UDG, in which the nodes are given by points in $\mathbb{R}^2$, and two nodes are connected by an edge if and only if one of the corresponding points is inside the unit-radius circle centered at the other point. We will use a capital Latin letter to represent a unit-radius circle, and we will use the same letter in bold face to describe the corresponding disk. For a unit-radius disk (circle) A, we will denote its center and the corresponding graph node by the same, but lower-case, letter a. Due to one-to-one correspondence between the nodes of $\mathcal{G}$ and the disks they represent, we can also refer to a disk $\mathbf{A}$ as the corresponding node of $\mathcal{G}$ whenever this simplifies the presentation. For a set of points $S \subseteq \mathbb{R}^2$, we will denote by $\mathcal{V}(S) = \{a \in \mathcal{V} : a \in S\}$ the subset of nodes of $\mathcal{G}$ that are given by points from S.

Assume that a pair of circles A and B intersect in exactly two points P_1 and P_2, i.e., $A \cap B = \{P_1, P_2\}$. Then the intersection $\mathbf{A} \cap \mathbf{B}$ of the corresponding pair of disks is called a *lens*, and P_1, P_2 are the *vertices* of this lens. Consider a set of three distinct pairwise overlapping disks $\mathbf{A}, \mathbf{B}$ and $\mathbf{C}$. We will call this set *triangular* if either (1) $S = \mathbf{A} \cap \mathbf{B} \cap \mathbf{C} \neq \emptyset$ and the boundary of S consists of three circular arcs, each belonging to a different circle; or (2) $\mathbf{A} \cap \mathbf{B} \cap \mathbf{C} = \emptyset$ and there is a nonempty area S with boundary consisting of three circular arcs, one from each circle, that has no points in the interior belonging to any of the disks (see Fig. 5.2). Then we call S a *convex circular triangle* in the first case and a *concave circular triangle* in the second case, respectively. Let $A \cap B = \{P_1, X\}, A \cap C = \{P_2, Y\}$, and $B \cap C = \{P_3, Z\}$, where $\{P_1, P_2, P_3\} \subset S$ are the "corner" points of S. We call P_1, P_2, and P_3 the *vertices* of the circular triangle and use the notation $\triangle P_1P_2P_3$ to represent a circular triangle in both cases. A triangular set of disks defines one circular triangle (convex or concave) and three lenses. The vertices P_1, P_2, P_3 of the circular triangle are also vertices of the corresponding lenses. We will call these three lens vertices the *inner vertices*, while the remaining three lens vertices, X, Y, Z, will be referred to as *outer vertices*. If $P_1 = P_2 = P_3$, i.e., A, B, and C all intersect in one point, Johnson's circle theorem [56] claims that the triangles $\triangle XYZ$ and $\triangle abc$ are similar.

Consider a set S of points on the plane overlapped by multiple disks, and suppose the boundary of S is formed by a set of k different circles. We will refer to these circles as *border circles*, as in Fig. 5.3 (left). Note that in the lens $\mathbf{A} \cap \mathbf{B}$ corresponding to two arbitrary border circles A and B,

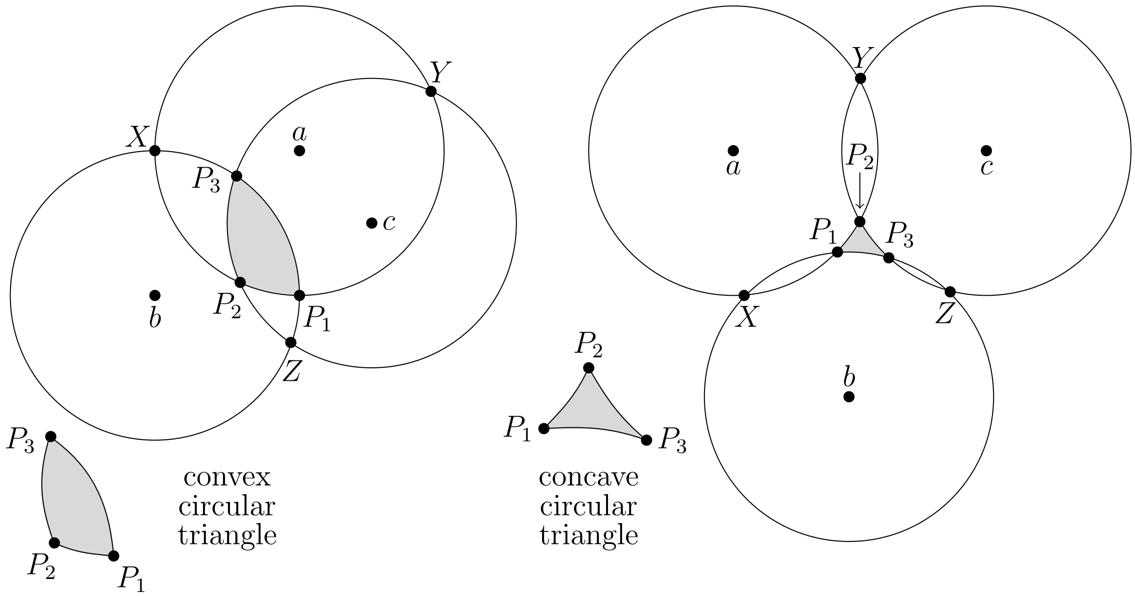


Figure 5.2: Circular triangles.

the set $\mathbf{A} \cap \mathbf{B} \setminus S$ will consist of at most two continuous regions, S' and S'' (see Fig. 5.3, right). Helly's theorem in two dimensions [28] states that if $\mathcal{F}$ is a finite family of at least 3 convex sets on the 2-dimensional plane and every 3 members of $\mathcal{F}$ have a common point, then there is a point common to all members of $\mathcal{F}$.

Next we review some terminology and notation from graph theory. Given a simple undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, if $(u, v) \in \mathcal{E}$ we call u and v *adjacent* or *neighbors*, and say that u and v *dominate* each other. Let $N_{\mathcal{G}}[v] = \{v\} \cup \{u \in \mathcal{V} : (u, v) \in \mathcal{E}\}$ be the *closed neighborhood* of v in $\mathcal{G}$. We call a subset $\mathcal{S}$ of nodes k-dominated in $\mathcal{G}$ if there is a subset $\mathcal{D} \subseteq \mathcal{V}$ of at most k nodes such that for any $u \in \mathcal{S} \setminus \mathcal{D}$ there is $v \in \mathcal{D}$ such that $(u, v) \in \mathcal{E}$. The subgraph induced by a subset of nodes $\mathcal{S}$ is denoted by $\mathcal{G}[\mathcal{S}]$. We denote by $d_{\mathcal{G}}(u, v)$ the distance between $u, v \in \mathcal{V}$ in $\mathcal{G}$ and by $diam(\mathcal{G})$ the graph-theoretic diameter of $\mathcal{G}$. A subset of nodes $\mathcal{C}$ is called a clique if for any $u, v \in \mathcal{C}$ we have $(u, v) \in \mathcal{E}$. For a positive integer s, a subset of nodes $\mathcal{K}$ is called an s-clique if for any $u, v \in \mathcal{K}$ we have $d_{\mathcal{G}}(u, v) \leq s$. An s-clique $\mathcal{K}$ is called an s-club if $diam(\mathcal{G}[\mathcal{K}]) \leq s$. Note that for $s = 1$ both s-clique and s-club become a clique. In this paper we consider the case of $s = 2$. For a positive integer t, the t^{th} power $\mathcal{G}^t$ of graph $\mathcal{G}$ is given by $\mathcal{G}^t = (\mathcal{V}, \mathcal{E}^t)$, where $\mathcal{E}^t = \{(u, v) : d_{\mathcal{G}}(u, v) \leq t\}$. Clearly, $\mathcal{K}$ is an s-clique in $\mathcal{G}$ if and only if $\mathcal{K}$ is a clique in $\mathcal{G}^s$.

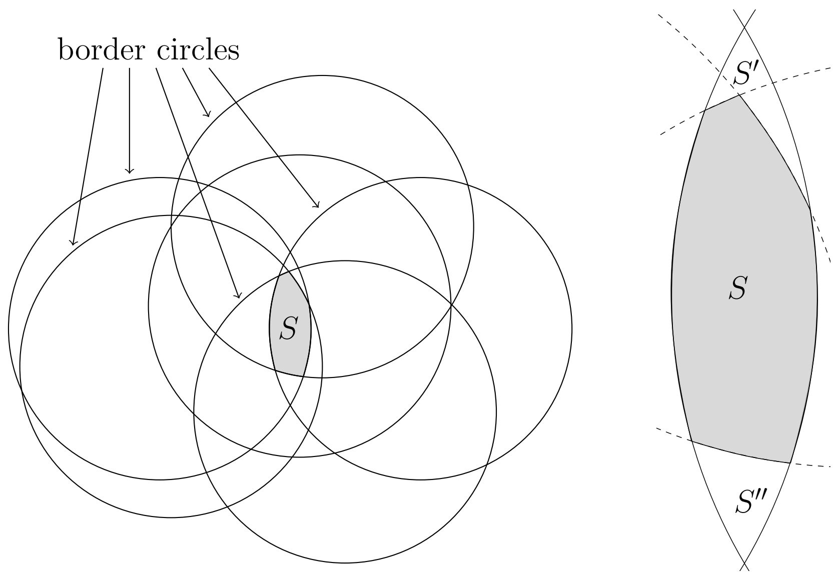


Figure 5.3: Border circles and a lens corresponding to two border circles.

5.3 Domination of 2-Cliques in Unit Disk Graphs

In this section we show that all 2-cliques in a UDG are 4-dominated, a result that is at the core of the proposed approximation algorithm. Note that this is not true for 2-cliques in general graphs. It is shown in [67] how to construct graphs of diameter 2 with minimum dominating set exceeding any size k.

Proposition 10. *Any 2-clique in a UDG is 4-dominated.*

Proof. Let $\mathcal{K}$ be an arbitrary 2-clique in a UDG $\mathcal{G}$. A key detail to note is that we do not require the elements in a dominating set for $\mathcal{K}$ to be members of $\mathcal{K}$. Since we are working with the containment model of UDG, every pair of disks $\mathbf{A}$ and $\mathbf{B}$ in the 2-clique $\mathcal{K}$ must intersect and there must be a node of $\mathcal{G}$ in $\mathbf{A} \cap \mathbf{B}$ to ensure that $d_{\mathcal{G}}(a, b) \leq 2$. We break our proof down into two cases. In the first case we assume that there exist three disks $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ in $\mathcal{K}$ that intersect pairwise but $\mathbf{A} \cap \mathbf{B} \cap \mathbf{C} = \emptyset$. In the other case we have $\mathbf{A} \cap \mathbf{B} \cap \mathbf{C} \neq \emptyset$ for any three disks $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ in $\mathcal{K}$.

Case 1: There exist $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ in $\mathcal{K}$ such that $\mathbf{A} \cap \mathbf{B} \cap \mathbf{C} = \emptyset$. Consider three disks $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ in $\mathcal{K}$ that yield a concave circular triangle with the largest area. We prove in Lemmas 19 and 21 (in subsection 5.3.1) that every other disk in $\mathcal{K}$ must overlap at least one entire lens $\mathbf{A} \cap \mathbf{B}$, $\mathbf{A} \cap \mathbf{C}$, or $\mathbf{B} \cap \mathbf{C}$. Since each of these lenses must contain a node of $\mathcal{G}$ for $\mathbf{A}$, $\mathbf{B}$, and

$\mathbf{C}$ to be in a 2-clique, taking one node from each lens provides three nodes dominating the entire 2-clique. Thus, $\mathcal{K}$ is 3-dominated in this case.

Case 2: $\mathbf{A} \cap \mathbf{B} \cap \mathbf{C} \neq \emptyset$ for any $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ in $\mathcal{K}$. By Helly's theorem there exists a set of points S in $\mathbb{R}$ that are common for all members of $\mathcal{K}$ (see Fig. 5.3, left). Clearly, if there is a node of $\mathcal{G}$ in S, the 2-clique is 1-dominated. Assume that S contains no nodes of $\mathcal{G}$. Then we choose an arbitrary pair of border disks $\mathbf{A}$ and $\mathbf{B}$ in $\mathcal{K}$ corresponding to non-consecutive pieces of the border of S and consider their intersection. Since $\mathbf{A}$ and $\mathbf{B}$ belong to the 2-clique $\mathcal{K}$, there must be a node p of $\mathcal{G}$ in $\mathbf{A} \cap \mathbf{B} \setminus S$. Further, since $\mathbf{A}$ and $\mathbf{B}$ are border disks for S that do not define consecutive border pieces, S will divide $\mathbf{A} \cap \mathbf{B}$ into two parts, S' and S'', as in Fig. 5.3 (right). If both parts contain the graph's nodes, p' and p'', respectively, then it is impossible to insert a disk between these two points without changing the border of S, which would be a contradiction since all disks must overlap S entirely. This implies that the 2-clique is 2-dominated by p' and p''. Thus, we only need to consider the case where only one of the sets S', S'' contains a node p of the graph. Without loss of generality (WLOG), suppose all such nodes lie in S'. In this case, we will use a finite sequence of steps to identify three border disk of S in $\mathcal{K}$, such that no node of the graph belonging to S' lies inside their intersection. Let $C_1, C_2, \ldots, C_k$ be the circles defining the common border of S and S' listed in the order in which the corresponding pieces appear as the common border $S \cap S'$ is navigated from A to B. Let $C_0 = A$. Starting with $i = 1$, we consider the intersection $\mathbf{A} \cap \mathbf{B} \cap \mathbf{C}_i$. If $\mathbf{A} \cap \mathbf{B} \cap \mathbf{C}_i$ does not contain a node of G, we stop. Otherwise, noting that $\mathcal{V}(\mathbf{B} \cap \mathbf{C}_i) \subset \mathcal{V}(\mathbf{B} \cap \mathbf{C}_{i-1}) \subset S'$, we set $A = C_i$ and restart the procedure for the new pair $\mathbf{A}, \mathbf{B}$. Since every repetition reduces the number of border disks between $\mathbf{A}$ and $\mathbf{B}$ by one, we eventually will produce two border disks $\mathbf{A}$ and $\mathbf{B}$ close enough together on the border of S such that $\mathcal{V}(\mathbf{A} \cap \mathbf{B}) \cap S' \neq \emptyset$, but $\mathcal{V}(\mathbf{C}_i \cap \mathbf{A} \cap \mathbf{B} \cap S') = \emptyset$. Since $\mathcal{V}(\mathbf{C}_i \cap \mathbf{A} \cap \mathbf{B}) \subset S'$, this implies that $\mathcal{V}(\mathbf{C}_i \cap \mathbf{A} \cap \mathbf{B}) = \emptyset$. We have produced three border disks $\mathbf{A}, \mathbf{B}$ and $\mathbf{C} = \mathbf{C}_i$ such that $\mathbf{A} \cap \mathbf{B} \cap \mathbf{C}$ contains no nodes of $\mathcal{G}$. Thus, each of $\mathbf{A} \cap \mathbf{B}$, $\mathbf{A} \cap \mathbf{C}$, and $\mathbf{B} \cap \mathbf{C}$ must contain a different node of the graph. Let v_1, v_2, and v_3 be arbitrary nodes of $\mathcal{G}$ belonging to $\mathbf{A} \cap \mathbf{B}$, $\mathbf{A} \cap \mathbf{C}$, and $\mathbf{B} \cap \mathbf{C}$, respectively. Note that $\mathbf{V}_1$, $\mathbf{V}_2$ and $\mathbf{V}_3$ intersect pairwise, since a, b, and c belong to the corresponding lenses. Thus, they produce either a convex circular triangle Δ' or a concave circular triangle Δ'', depending on whether all three discs have points in common. In either

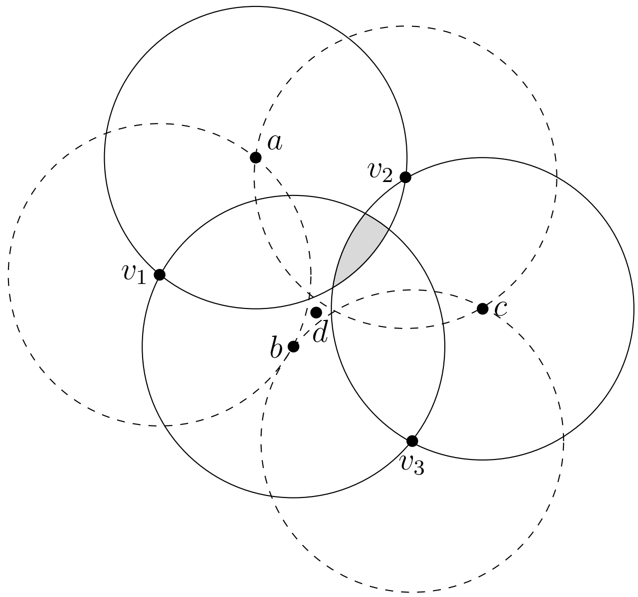


Figure 5.4: An illustration to Case 2 of the proof of Proposition 10.

case, $\mathcal{K}$ cannot contain a node corresponding to a point outside $\mathbf{V}_1 \cup \mathbf{V}_2 \cup \mathbf{V_3} \cup \Delta''$ since this would imply that one of the circles A, B, or C is not a border circle of S (see Fig. 5.4). Thus, if $\mathbf{V}_1 \cap \mathbf{V}_2 \cap \mathbf{V_3} \neq \emptyset$ then $\{v_1, v_2, v_3\}$ dominates $\mathcal{K}$ and the 2-clique is 3-dominated. It remains to consider the case where $\mathbf{V}_1 \cap \mathbf{V}_2 \cap \mathbf{V_3} = \emptyset$. Then $\Delta'' \neq \emptyset$ and it is possible that there is a node d in $\mathcal{K}$ that belongs to Δ'' (see Fig. 5.4). Then we show in Lemma 22 (in subsection 5.3.2) that one of $\mathbf{A}$, $\mathbf{B}$, or $\mathbf{C}$, whose centers belong to $\mathbf{V}_1 \cap \mathbf{V}_2$, $\mathbf{V}_1 \cap \mathbf{V}_3$, and $\mathbf{V}_2 \cap \mathbf{V}_3$, respectively, must cover the entire Δ'' and hence the proof is complete.

□

Corollary 4. *Any 2-club in a UDG is 3-dominated.*

Proof. Note that in Proposition 10, we could not conclude that 2-cliques are 3-dominated only in the the Case 2 of the proof, where $\mathbf{V}_1 \cap \mathbf{V}_2 \cap \mathbf{V_3} = \emptyset$. However, by definition of a 2-club, v_1, v_2, and v_3 must be in the 2-club, which is in contrast to 2-cliques. Thus, we are in fact in Case 1, for which we have already proven 3-domination. Thus, 2-clubs are always 3-dominated. □

Note that the fact that all 2-clubs are 3-dominated can potentially be used in designing exact algorithms for the maximum 2-club problem as follows. Instead of solving the problem for the original graph, we can solve it for induced subgraphs of all subsets of 3 vertices together with their neighbors. This may help solving instances where all such subgraphs are substantially smaller than the original graph.

Above, we established the upper bound of 4 on the minimum size of a dominating set for a 2-clique in a UDG. While producing an example of a 2-clique that is not 3-dominated appears to be difficult, if not impossible, we can show that our upper bound is nearly tight by providing an example of a 2-clique that is not 2-dominated. To do so, we adapt the example discussed in [22]. This example is similar to the one in [46], where it is proven that for a set of congruent disks that intersect pairwise, the piercing number, which is the fewest points in space that intersect every object in a given set, is precisely 3.

Example 1. *An example of a 3-dominated 2-clique in a UDG.*

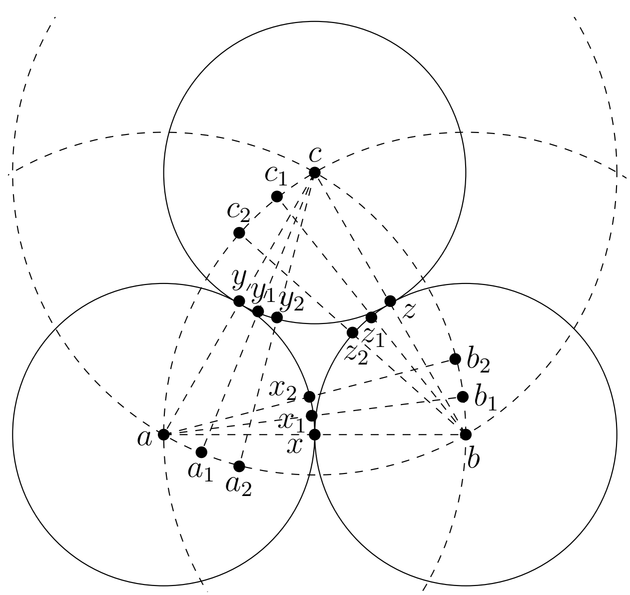


Figure 5.5: A 2-clique with a minimum dominating set of 3 vertices.

To construct the required 2-clique, which is also a 2-club, we consider a set of $3k$ unit disks $\{\mathbf{A}_i, \mathbf{B}_i, \mathbf{C}_i\}_{i=0}^{k-1}$, where $\mathbf{B} \equiv \mathbf{B}_0$, and $\mathbf{C} \equiv \mathbf{C}_0$. We position the centers of disks $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ at distance 2 away from each other. Let A', B', and C' be circles of radius 2 centered at a, b, and c, respectively. We position nodes $a_i \in \mathbf{A} \cap C'$, $b_i \in \mathbf{B} \cap A'$, and $c_i \in \mathbf{C} \cap B'$, $i = 1, \ldots, k-1$ and $x_i \in A, z_i \in B, y_i \in C, i = 0, \ldots, k-1$, where $x \equiv x_0, y \equiv y_0, z \equiv z_0$, as shown in Fig. 5.5. Note that for $i = 0, \ldots, k-1$, x_i, y_i, and z_i are midpoints between a and b_i, c and a_i, and b and c_i, respectively. Let $\mathcal{V}_k = \{a_i, b_i, c_i, x_i, y_i, z_i : i = 0, \ldots, k-1\}$, then $|\mathcal{V}_k| = 6k$ and no two disks contain more than $5k$ nodes combined, while $\mathcal{V}_k$ is dominated by $\{a, b, c\}$. Thus, $\mathcal{V}_k$ is a 3-dominated 2-clique that is not 2-dominated.

5.3.1 When $\exists A, B, C \in \mathcal{K}$ s.t. $A \cap B \cap C = \emptyset$

In this part, we prove several lemmas about the case that $A \cap B \cap C = \emptyset$.

Lemma 17. *Let A, B and C be circles of equal radius r with the centers a, b and c, respectively. Assume that $\triangle abc$ is acute, and has circumradius $\rho > 0$. Let $A \cap B = \{P_1, P_1'\}$, $A \cap C = \{P_2, P_2'\}$, and $B \cap C = \{P_3, P_3'\}$ be the intersection points of the pairs of circles, with P_1', P_2', and P_3' being the three outer vertices of the corresponding lenses, and let $\delta = sign(r^2 - \rho^2)$. Let ρ' be the circumradius of $\triangle P_1'P_2'P_3'$ and let ρ'' be the circumradius of $\triangle P_1P_2P_3$. Then*

$$0 \le \rho', \rho'' \le r \le \rho \le \rho' + \rho'' \quad \textit{if} \quad \delta < 0$$
$$0 \le \rho, \rho'' \le r \le \rho' \le \rho + \rho'' \quad \textit{if} \quad \delta > 0.$$

Proof. This is an adaptation of Theorems 3 and 4 in [68]. □

Lemma 18. *If three disks $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ overlap pairwise but $\mathbf{A} \cap \mathbf{B} \cap \mathbf{C} = \emptyset$, then the triangle formed by connecting their centers is acute. Moreover, all points of each circle that belong to another disk are located on its arc with the central angle less than π.*

Proof. Consider the midpoint m of side bc, which lies in $\mathbf{B} \cap \mathbf{C}$ as in Fig. 5.6 (left). Since $\mathbf{A} \cap \mathbf{B} \cap \mathbf{C} = \emptyset$, it must be that $|am| > 1$. Consider a circle $\bar{M}$ of radius $|mc|$ centered at m. Then a cannot be inside this circle. Consider the point Q of intersection of the side ab with $\bar{M}$. Then angle $\angle bQc$ is a right angle. But this means $\angle bac$ is acute because triangle $\triangle cQa$ has right angle $\angle cQa$. By similar arguments we can conclude that angles $\angle bca$ and $\angle abc$ are acute and hence $\triangle abc$ is acute.

To prove the second statement, consider any of the three circles, e.g., B. All points that this circle shares with $\mathbf{A}$ or $\mathbf{C}$ are located on its arc between X and Z, hence it suffices to show that $\angle XbZ < \pi$. We have $\angle XbZ = 2\angle QbP_1 + \angle P_1bP_3 + 2\angle P_3bm \le 2\angle abc < \pi$ since $\angle abc$ is acute.

□

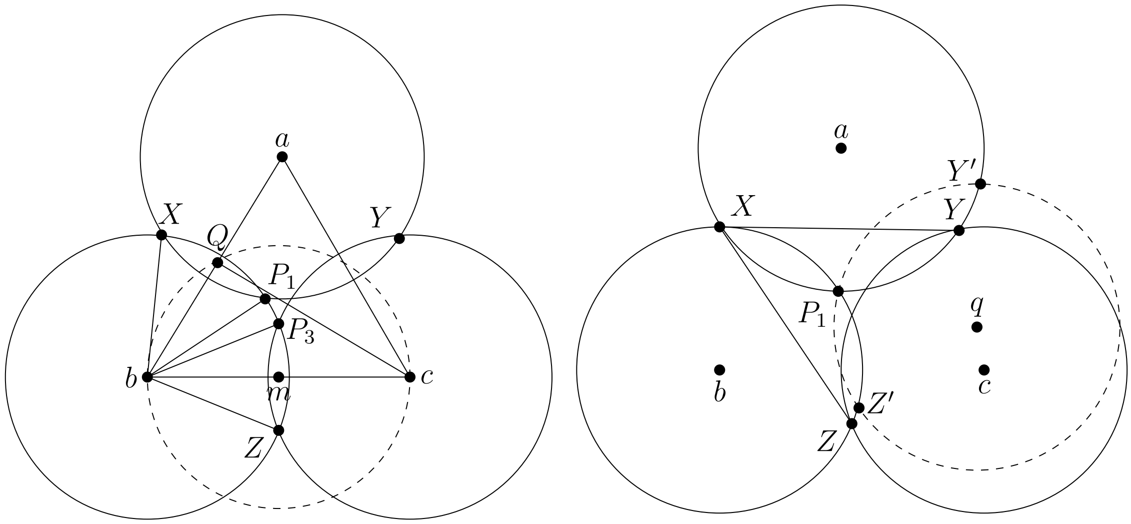


Figure 5.6: Illustrations to the proofs of Lemmas 18 and 20.

Lemma 19. *Let* $\mathbf{A}$, $\mathbf{B}$, *and* $\mathbf{C}$ *be three disks centered at three nodes of 2-clique* $\mathcal{K}$ *with* $\mathbf{A} \cap \mathbf{B} \cap \mathbf{C} = \emptyset$ *that produce a concave circular triangle* $\triangle P_1P_2P_3$ *of area* σ. *Then any other disk* $\mathbf{Q}$ *corresponding to* $q \in \mathcal{K}$ *that does not contain* $\triangle P_1P_2P_3$ *and does not form a concave circular triangle of area larger than* σ *with any two of the disks* $\mathbf{A}, \mathbf{B}, \mathbf{C}$ *must contain one of the lenses* $\mathbf{A} \cap \mathbf{B}$, $\mathbf{A} \cap \mathbf{C}$, *or* $\mathbf{B} \cap \mathbf{C}$ *entirely.*

Proof. Since the concave circular triangle $\triangle P_1P_2P_3$ has larger area than a concave circular triangle that $\mathbf{Q}$ forms with any two of the disks $\mathbf{A}, \mathbf{B}, \mathbf{C}$, the circle Q must intersect the boundary of $\triangle P_1P_2P_3$. Then there is at least one vertex among P_1, P_2, and P_3 that is not in $\mathbf{Q}$. There are two possible subcases: only one of the vertices P_1, P_2, P_3 is not in $\mathbf{Q}$ (subcase 1); and two of the vertices P_1, P_2, P_3 are not in $\mathbf{Q}$ (subcase 2).

Subcase 1: Let the vertex not in $\mathbf{Q}$ be P_2. Since $q \in \mathcal{K}$, $\mathbf{Q}$ must have a nonempty overlap with $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$. Let X_1 and X_2 denote the intersections of circles Q and A, and Y_1 and Y_2 denote the intersections of circles Q and C. If $\mathbf{Q}$ contains neither $\mathbf{A} \cap \mathbf{B}$ nor $\mathbf{B} \cap \mathbf{C}$, then one of the points of intersection of circles Q and A (WLOG it is X_1) must lie in $\mathbf{A} \cap \mathbf{B}$, and one of the points of intersection of circles Q and C (let it be Y_1) must lie in $\mathbf{B} \cap \mathbf{C}$. Also because circle Q passes through $X_1 \in \mathbf{A} \cap \mathbf{B}, Y_1 \in \mathbf{B} \cap \mathbf{C}$ and must intersect P_1P_2 or P_2P_3, it must intersect B at two points Z_1 and Z_2 inside the circular arc XP_1Z (as in Fig. 5.7,

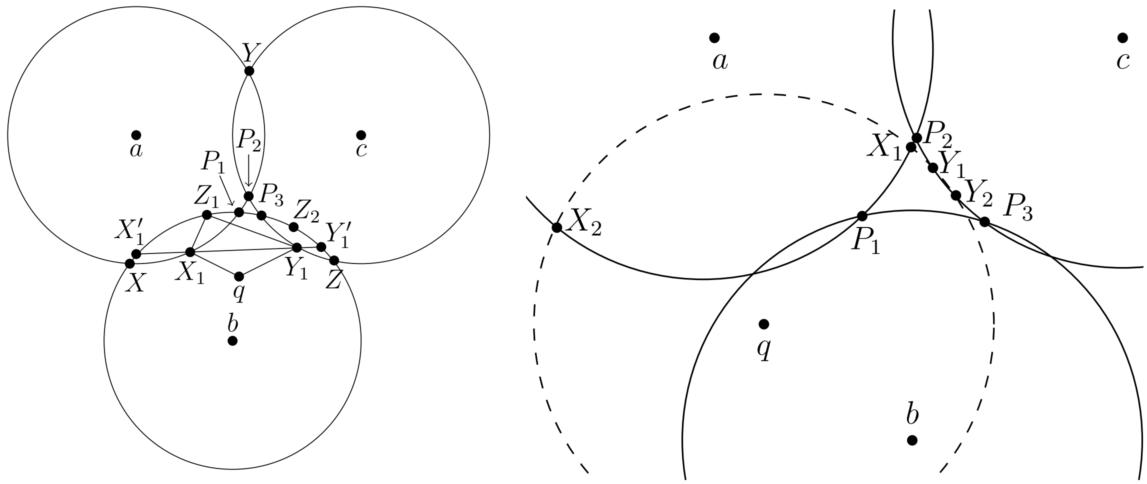


Figure 5.7: An illustration to the proof of Lemma 19.

left). Let X_1' and Y_1', respectively, be the points of intersection of the line passing through X_1 and Y_1 with the circle B. Then, the circle B circumscribes the triangle $X_1'Z_1Y_1'$, the circle Q circumscribes the triangle $X_1Z_1Y_1$, and since both circles are of same diameter, we have $\frac{|X_1'Y_1'|}{\sin(\angle X_1'Z_1Y_1')} = \frac{|X_1Y_1|}{\sin(\angle X_1Z_1Y_1)}$. Since $|X_1'Y_1'| > |X_1Y_1|$ and $\angle X_1'Z_1Y_1' > \angle X_1Z_1Y_1$, the angle $\angle X_1Z_1Y_1$ must be acute. Then the angle $\angle X_1qY_1$ corresponding to the circular arc containing Z_1 is $\angle X_1qY_1 = 2(\pi - \angle X_1Z_1Y_1) > \pi$. Since $X_1 \in A, Y_1 \in C$, this contradicts the second statement of Lemma 18 applied to circles A, C, and Q. Thus, $\mathbf{Q}$ contains at least one of the lenses $\mathbf{A} \cap \mathbf{B}$, $\mathbf{B} \cap \mathbf{C}$, and the proof of subcase 1 is complete.

Subcase 2: Let the vertices not in $\mathbf{Q}$ be P_2 and P_3 (see Fig. 5.7, right). Then rotate $\mathbf{Q}$ around X_1 so that Y_2 coincides with P_3. Let $\mathbf{Q'}$ be the disk obtained from $\mathbf{Q}$ as the result of this rotation. Then we arrive at subcase 1 implying that $\mathbf{Q'}$ contains $\mathbf{A} \cap \mathbf{B}$. Noting that $\mathbf{Q'} \cap \mathbf{A} \subseteq \mathbf{Q} \cap \mathbf{A}$, we conclude that $\mathbf{Q}$ contains $\mathbf{A} \cap \mathbf{B}$.

□

Lemma 20. *Suppose three disks* $\mathbf{A}$, $\mathbf{B}$, *and* $\mathbf{C}$ *overlap pairwise but* $\mathbf{A} \cap \mathbf{B} \cap \mathbf{C} = \emptyset$. *Let* X, Y, *and* Z *be the outer vertices of the lenses* $\mathbf{A} \cap \mathbf{B}$, $\mathbf{A} \cap \mathbf{C}$, *and* $\mathbf{B} \cap \mathbf{C}$, *respectively. Then* $\triangle XYZ$ *is acute.*

Proof. Draw another circle Q as in Fig. 5.6 (right) that passes through the inner vertex of $\mathbf{A} \cap \mathbf{B}$, so that $A \cap B \cap Q = \{P_1\}$, and through a point $Z' \neq Z$ that lies on the circle B and belongs to $\mathbf{B} \cap \mathbf{C}$.

Note that we can select such Z' arbitrarily close to Z so that $\epsilon = \angle Z'XZ$ can be chosen as small as needed. By Lemma 19, since $\mathbf{Q}$ cannot overlap $\mathbf{B} \cap \mathbf{C}$, it must be that $\mathbf{Q}$ contains $\mathbf{A} \cap \mathbf{C}$. Thus Y is inside $\mathbf{Q}$ and $\angle ZXY \leq \angle ZXY' = \angle Z'XY' + \epsilon$. By Johnson's circle theorem [56], $\triangle Z'XY'$ is similar to the triangle made of the centers of the three circles, which is acute by Lemma 18. Thus $\angle Z'XY'$ is acute and since ϵ can be made arbitrarily small, $\angle ZXY$ is also acute. We can use the same argument to conclude $\angle ZYX$ and $\angle XZY$ are also acute, so the triangle is acute. □

Lemma 21. *Let disks* $\mathbf{A}$, $\mathbf{B}$, *and* $\mathbf{C}$, *where* $\mathbf{A} \cap \mathbf{B} \cap \mathbf{C} = \emptyset$, *be nodes of 2-clique* $\mathcal{K}$ *that produce a concave circular triangle* $\triangle P_1P_2P_3$. *Then any other disk* $\mathbf{Q}$ *in* $\mathcal{K}$ *that overlaps* $\triangle P_1P_2P_3$ *entirely must contain one of the lenses* $\mathbf{A} \cap \mathbf{B}$, $\mathbf{A} \cap \mathbf{C}$, *or* $\mathbf{B} \cap \mathbf{C}$.

Proof. Suppose by way of contradiction that a disk $\mathbf{Q}$ could contain the concave circular triangle and not overlap a lens entirely. Then Q must intersect the border of each lens at two points. Call these points J, K, M, N, R, and S, as in Fig. 5.8 (left). From the figure, if $\mathbf{Q}$ is to overlap the entire

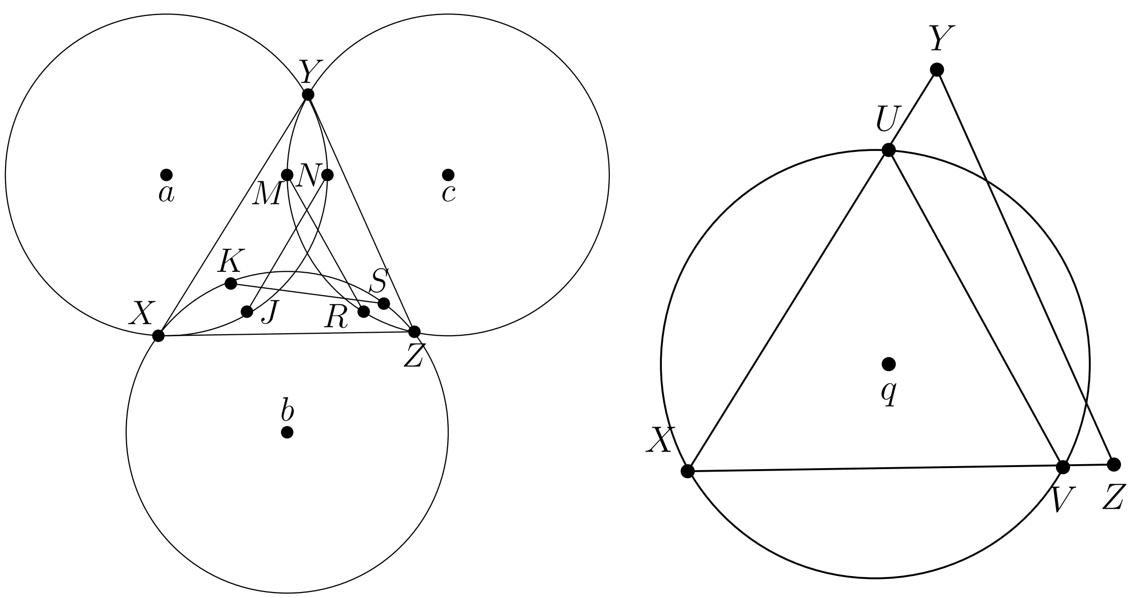


Figure 5.8: An illustration for the proof of Lemma 21.

concave circular triangle, its center q and a must be on the opposite sides of the line passing through J and N. Similarly q and b (q and c) must be separated by the line passing through K and S (R and

M). This is sufficient to ensure that q must be strictly within the triangle $\triangle XYZ$. If any of X, Y, or Z is inside $\mathbf{Q}$ then $\mathbf{Q}$ covers an entire lens and we have contradicted our assumption above. Thus it must be that $|qX| > 1$, $|qY| > 1$, and $|qZ| > 1$. Suppose WLOG the minimum of $\{|qX|, |qY|, |qZ|\}$ is $|qX|$. If we draw a circle centered at q with radius $|qX|$, then Y and Z must still be outside this circle. Thus this circle must have intersection points with both XY and XZ, which we will call U and V, respectively. Since $\triangle XYZ$ is acute by Lemma 20, we have $|UV| < |YZ|$ (this can be easily shown by, e.g., applying the cosine law to compute $|UV|$ and $|YZ|$). Let ρ be the radius of the circumcircle through X, Y, and Z, then $\rho = \frac{|YZ|}{2\sin(\angle YXZ)} > \frac{|UV|}{2\sin(\angle YXZ)} = \frac{|UV|}{2\sin(\angle UXV)} = |qX| > 1$. Let τ be the radius of the circle through a, b, and c. Clearly $\tau > 1$ since $\mathbf{A} \cap \mathbf{B} \cap \mathbf{C} = \emptyset$. Then Lemma 17 says that since $\delta = \text{sign}(1 - \tau^2) < 0$, it must be that $0 \leq \rho \leq 1$, where ρ is the circumradius of the circle passing through the outer intersections X, Y and Z of circles A, B, and C. But this is a contradiction to $\rho > 1$ above and thus it must be impossible that a circle Q exists as described.

□

5.3.2 When $A \cap B \cap C \neq \emptyset$, $\forall A, B, C \in \mathcal{K}$

In this part we prove a lemma used in the case that $A \cap B \cap C \neq \emptyset$

Lemma 22. *Suppose $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ are three disks that intersect pairwise, but $\mathbf{A} \cap \mathbf{B} \cap \mathbf{C} = \emptyset$, so that they define a concave circular triangle $\triangle P_1P_2P_3$. Suppose WLOG that $\mathbf{B} \cap \mathbf{C}$ has area less than or equal to that of $\mathbf{A} \cap \mathbf{B}$ and $\mathbf{A} \cap \mathbf{C}$. Then a unit-radius circle D centered at any $d \in \mathbf{B} \cap \mathbf{C}$ will cover the entire circular triangle $\triangle P_1P_2P_3$.*

Proof. It is sufficient to show the circle D centered at the outer vertex d of $\mathbf{B} \cap \mathbf{C}$ covers the vertices of the circular triangle, specifically P_1 and P_2 since they are most distant from d. Assume WLOG that $\mathbf{A} \cap \mathbf{B}$ has smaller area than $\mathbf{A} \cap \mathbf{C}$, in which case we rotate circle B around P_1 to create a new circle B' such that $\mathbf{A} \cap \mathbf{B'}$ and $\mathbf{C} \cap \mathbf{B'}$ are equal, as in Fig. 5.9. It is clear that $\mathbf{A} \cap \mathbf{B'} \subseteq \mathbf{A} \cap \mathbf{B}$. Let d' be the intersection of B' and C, and let d'' be the point on B' obtained from d as a result of the rotation. Since d is on arc P_2d' of C, it is clear that $|P_2d'| \geq |P_2d|$. Since $|P_1d| = |P_1d''|$ and d'' is on arc P_1d' of B', we have $|P_1d'| \geq |P_1d|$. Also $|ab'| = |cb'| \geq |ac|$ because the area of lens $\mathbf{A} \cap \mathbf{C}$ is larger than that of lens $\mathbf{A} \cap \mathbf{B}$, which is larger than the area of lens $\mathbf{A} \cap \mathbf{B'}$.

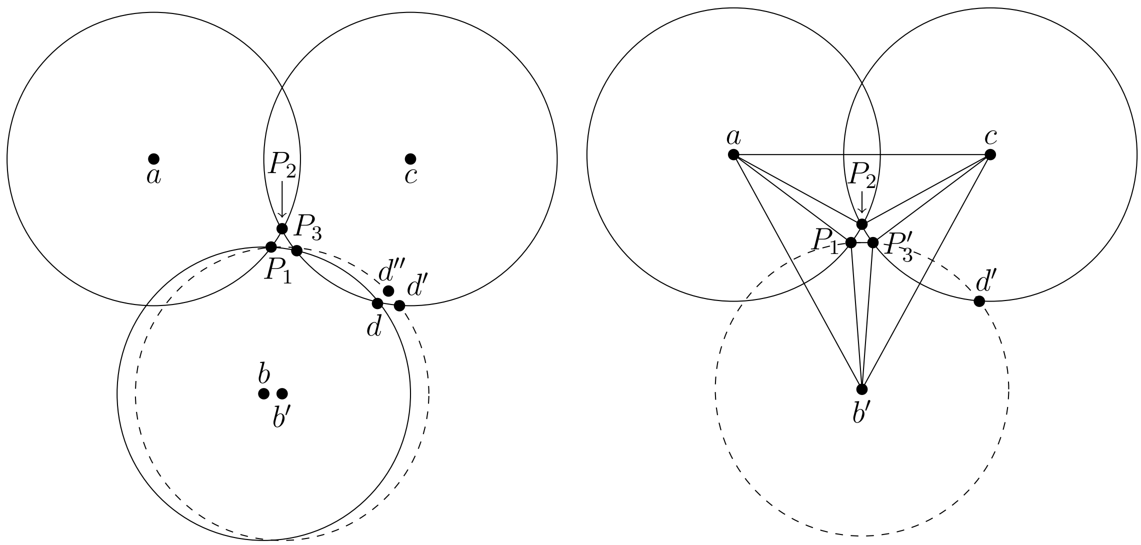


Figure 5.9: An illustration to the proof of Lemma 22.

Note that $|aP_2| = |cP_2| = |cP_3'| = |b'P_3'| = |b'P_1| = |aP_1| = 1$. Let $\alpha = \angle P_2ac = \angle P_2ca$, $\beta = \angle P_2aP_1 = \angle P_2cP_3'$, $\gamma = \angle P_1ab' = \angle P_1b'a = \angle P_3'b'c = \angle P_3'cb'$, and $\delta = \angle P_1b'P_3'$.

The key to the proof is showing that $\beta \leq \delta$. First note that $\gamma \leq \alpha$ since α corresponds to half-arc of the lens $\mathbf{A}\cap\mathbf{C}$ and γ corresponds to the half-arc of the lens $\mathbf{B}'\cap\mathbf{C}$, which is not larger than $\mathbf{A}\cap\mathbf{C}$. Next, suppose for contradiction that $\delta < \beta$. Then $|P_1P_3'| < |P_1P_2| = |P_2P_3|$. But at the same time $\angle P_1P_2P_3' = 2\pi - \angle aP_2P_1 - \angle aP_2c - \angle cP_2P_3' = 2\alpha + \beta$. Similarly $\angle P_2P_3'P_1 = 2\gamma + \frac{\beta}{2} + \frac{\delta}{2}$. If $\delta < \beta$ then $\angle P_2P_3'P_1 = 2\gamma + \frac{\beta}{2} + \frac{\delta}{2} < 2\alpha + \beta = \angle P_1P_2P_3'$ since we know $\gamma \leq \alpha$. But $\angle P_2P_3'P_1 < \angle P_1P_2P_3'$ implies that $|P_1P_2| < |P_1P_3'|$, contradicting to the opposite inequality $|P_1P_3'| < |P_1P_2|$ established above. Hence, $\delta \geq \beta$.

Since $\beta \leq \delta$, we can conclude that $\angle d'cP_2 = 2\gamma + \beta \leq 2\gamma + \delta = \angle cb'a \leq \frac{\pi}{3}$ since $|ab'| = |cb'| \geq |ac|$. This implies that $|d'P_2| \leq |CP_2| = 1$. Also, $\angle d'b'P_1 = \angle cb'a \leq \frac{\pi}{3}$, implying $|d'P_1| \leq 1$. Thus, $|P_1d| \leq |P_1d'| \leq 1$, $|P_2d| \leq |P_2d'| \leq 1$, and the circle centered at d covers both P_1 and P_2.

□

5.4 Domination of Graphs with Diameter 2 in General Graphs

Proposition 10 implies that the minimum dominating set problem is polynomially solvable in UDGs of diameter 2. In this section we show that this is not the case, unless $\mathcal{P} = \mathcal{NP}$, for general diameter-two graphs. Let Dominating Set (DS) be the decision version of the minimum dominating set problem, which, given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and a positive integer $k \le |\mathcal{V}|$ asks if there is a subset $\mathcal{D}$ of k nodes in $\mathcal{G}$ such that each node not in $\mathcal{D}$ has at least one neighbor in $\mathcal{D}$.

Proposition 11. Dominating Set *remains NP-complete when restricted to graphs of diameter two.*

Proof. Let us denote by DS2 the DS problem restricted to graphs of diameter two. Clearly, DS2 is in NP. To show NP-completeness, we use a reduction from Vertex Cover (VC) problem, which, given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and a positive integer $k \le |\mathcal{V}|$ asks if there is a subset $\mathcal{C}$ of k nodes in $\mathcal{G}$ such that every edge in $\mathcal{E}$ has at least one endpoint in $\mathcal{C}$. VC is a classical NP-complete problem [35]. Given an instance of VC, we construct an instance $\mathcal{G}' = (\mathcal{V}', \mathcal{E}')$ of DS2 in polynomial time as follows. For each edge $(i, j) \in \mathcal{E}, i < j$, we introduce two distinct nodes v_{ij} and v_{ji} in $\mathcal{G}'$. Let $\mathcal{V}_1' = \{v_{ij}, v_{ji} : (i, j) \in \mathcal{E}, i < j\}$ be the set of all such nodes. In addition, for each two edges (i, j) and (p, q) with no common endpoint in $\mathcal{G}$ and $i < j, i < p < q$, we introduce four nodes $v_{ij}^{pq}, v_{ij}^{qp}, v_{ji}^{pq}, v_{ji}^{qp}$ in $\mathcal{G}'$. Let $\mathcal{V}_2' = \{v_{ij}^{pq}, v_{ij}^{qp}, v_{ji}^{pq}, v_{ji}^{qp} : (i, j), (p, q) \in \mathcal{E}, i < j, i < p < q, p \ne j \ne q\}$. Then the graph $\mathcal{G}' = (\mathcal{V}', \mathcal{E}')$ is given by

$$\mathcal{V}' = \mathcal{V} \cup \mathcal{V}_1' \cup \mathcal{V}_2'; \;\; \mathcal{E}' = \mathcal{E}_1' \cup \mathcal{E}_2' \cup \mathcal{E}_3',$$

where $\mathcal{E}_1' = \{(v', v'') : v', v'' \in \mathcal{V} \cup \mathcal{V}_2', v' \ne v''\}, \mathcal{E}_2' = \{(v, v_{ij}) : v \in \mathcal{V}, v_{ij} \in \mathcal{V}_1', v = i \text{ or } v = j\}$, and $\mathcal{E}_3' = \{(v_{rs}, v_{ij}^{pq}) : v_{rs} \in \mathcal{V}_1', v_{ij}^{pq} \in \mathcal{V}_2', (r = i \,\&\, s = j) \text{ or } (r = p \,\&\, s = q)\}$. The construction is illustrated in Figure 5.10. It is easy to check that $diam(\mathcal{G}') = 2$ and $\mathcal{G}$ has a vertex cover of size at most k if and only if $\mathcal{G}'$ has a dominating set of size at most k. $\square$

5.5 An Effective Algorithm to Find 2-Cliques in a Unit Disk Graph

Now we apply proposition 10 to develop an effective algorithm for 2-clique problem in a unit disk graph.

Proposition 12. *There exists a $\frac{1}{2}$-approximation algorithm for the maximum 2-clique problem in a UDG $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ that runs in $O(|\mathcal{V}|^{4.5})$ time.*

Proof. We claim that we can find the largest 2-clique dominated by 2 elements in any graph in polynomial time. First, for a pair $\{v_1, v_2\}$ of nodes consider a graph $\mathcal{G}'(v_1, v_2)$, which is the subgraph of $\mathcal{G}^2$ induced by the union of their closed neighborhoods $N_{\mathcal{G}}[v_1] \cup N_{\mathcal{G}}[v_2]$. Since both $N_{\mathcal{G}}[v_1]$ and

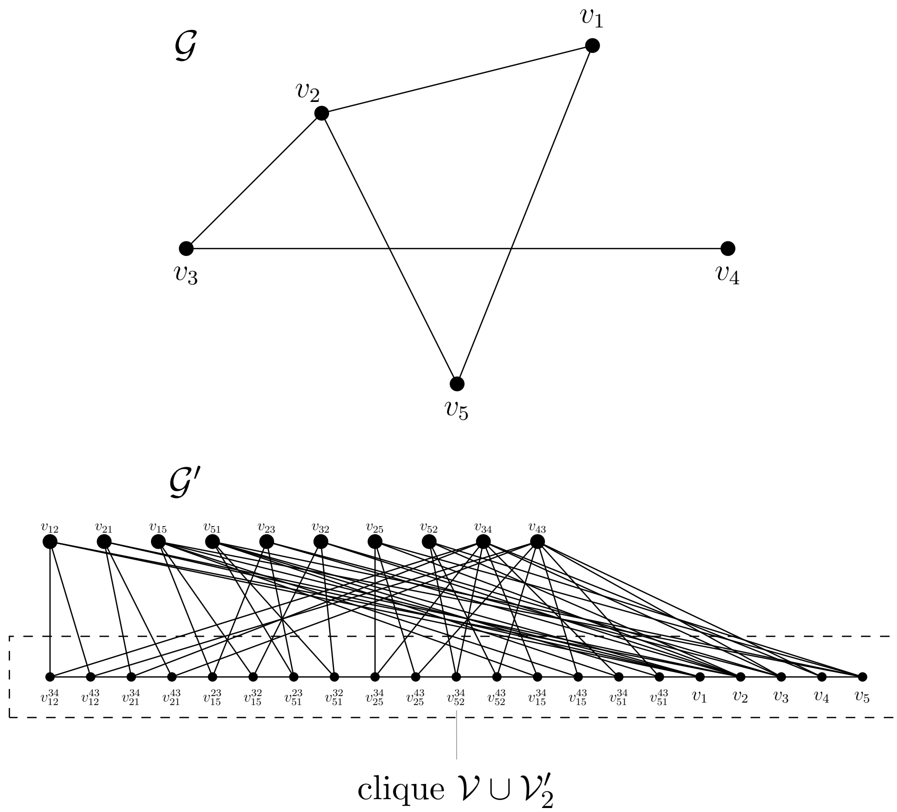


Figure 5.10: An illustration to the proof of Proposition 11.

$N_{\mathcal{G}}[v_2]$ are cliques in $\mathcal{G}^2$, $\mathcal{G}'(v_1, v_2)$ is a co-bipartite graph. Recall that the maximum clique problem can be solved in polynomial time on co-bipartite graphs [26]. Also, solving the maximum clique problem in $\mathcal{G}'(v_1, v_2)$ produces a maximum 2-clique in $\mathcal{G}$ that is dominated by v_1 and v_2. Hence, by considering all pairs of nodes corresponding to intersecting disks in $\mathcal{G}$, we can identify the largest 2-clique $\mathcal{K}'$ dominated by 2 elements in $\mathcal{G}$ in $O(|\mathcal{V}|^{4.5})$ time [26]. Also, since all 2-cliques are 4-dominated in a UDG by Proposition 10, at least half of the nodes of a maximum 2-clique $\mathcal{K}^*$ in $\mathcal{G}$ must be dominated by 2 nodes. Since a subset of a 2-clique of $\mathcal{G}$ is a 2-clique in the same graph, we conclude that $|\mathcal{K}'| \geq \frac{1}{2}|\mathcal{K}^*|$. $\square$

A uniform random UDG $\mathcal{G}(n, p)$ on n nodes is a UDG obtained by generating, uniformly and randomly, a set of n points within the square $[0, r] \times [0, r]$ in $\mathbb{R}^2$, where r is chosen such that an edge between a pair of points exists with probability p. The value of r can be determined by, e.g., using the results established in [83], where a formula for the probability distribution for Euclidean distance between random points in a box is established.

In a uniform random UDG, the approximation ratio of our algorithm can be better.

Proposition 13. *There exists an algorithm that finds, with asymptotic probability 1, a* $\frac{2}{3}$*-approximate solution to the maximum 2-clique problem in a uniform random UDG* $\mathcal{G}(n,p)$ *in* $O(n^{4.5})$ *time.*

Proof. As shown in [48], given a set of random points in a punctured unit disk, with asymptotic probability 1, there exist two points that will cover all the points in the unit disk (i.e., are Euclidean distance no more than 1 from any point in the unit disk). In the case where there is a nonempty set S overlapped by all disks (Case 2 in the proof of Proposition 10), we can take any point in S as the center of a punctured unit disk that will cover all points of the 2-clique, since all members of the 2-clique are within the circle of radius 1 of every point in S. In this case, with asymptotic probability 1, the set of disks in such a 2-clique is 2-dominated. In the other case, where $\mathbf{A} \cap \mathbf{B} \cap \mathbf{C} = \emptyset$ for any three disks (Case 1 in the proof of Proposition 10), we proved that the 2-clique is 3-dominated. Combining these results we conclude that our solution to the maximum 2-clique problem is, with asymptotic probability 1, a solution with $\frac{2}{3}$-approximation ratio to the 2-clique problem in uniform random UDGs. $\square$

While the above proposition can guarantee that the largest 2-clique in a uniform random UDG can be found with 2/3-approximation ratio with asymptotic probability 1, the algorithm performs even better in practice. In a sample set of experiments, we generated 3,500 uniform random UDGs of 50 nodes and 100 random UDGs of 100 nodes for each density in the range from .05 to 1 in increments of .05. In all 70,000 experiments with 50-node instances and all 2,000 experiments with 100-node instances, the size of the maximum 2-clique (computed using Östergård's algorithm [73] for the maximum clique in $\mathcal{G}^2$) and the 2-clique found by the proposed approximation algorithm matched.

6. THE TWO-GUARD PROBLEM IN CURVILINEAR POLYGONS

In this chapter, we generalize the *two-guard* problem from a simple polygon to a *piecewise locally-convex polygon*. We first describe the two-guard problem and the concept of piecewise locally convex polygons formally in Section 6.1. This problem asks whether such a polygon is *walkable*. Then we analyze the properties of piecewise locally convex polygons to solve the two-guard problem in Section 6.2. After that, we investigate necessary and sufficient conditions for a piecewise locally convex polygon to be walkable in analogy to the original two-guard problem in Section 6.3 and Section 6.4. Finally, we develop an algorithm to construct the solution when the polygon is walkable in Section 6.5.

6.1 Problem Description

In the previous chapter, we discussed the UDG model which defines connectivity via distance constraints. However, several other problems, like art gallery problems and guards problems, consider *visibility* as the connectivity requirement. Next, we define the concept of visibility formally.

Definition 2. *Given a set $P \in \mathbb{R}^n$, we say that a point $x \in P$ is visible from a point $y \in P$ if and only if the connecting line segment $\overline{xy}$ is entirely contained in P.*

The *two-guard* problem asks for a walk of two points (*guards*) on the boundary of a simple polygon P from the starting vertex s to the ending vertex t, one clockwise and one counterclockwise, such that the guards are always mutually visible.

Definition 3.

1. *A walk on P with s, t on the boundary of P is a pair (l, r) of continuous functions such that:*

   (a) $l : [0, 1] \to L, r : [0, 1] \to R,$

   (b) $l(0) = r(0) = s, l(1) = r(1) = g,$

   (c) *$l(t)$ is visible from $r(t)$ for all $t \in [0, 1]$.*

   Any line segment $\overline{l(t)r(t)}$ is called a walk line segment of the walk. The point $r(t)$ is the walk partner of $l(t)$, and vice versa.

2. *A walk on P from s to t is called straight if both l and r are non-decreasing with respect to the orientation of L and R.*

3. P from s to t is called walkable if it admits a straight walk.

TWO-GUARD: Given a polygon P and s, t on the boundary of P, is P walkable from s to t?

The two-guard problem is first introduced by Icking and Klein [53], who developed an $\mathcal{O}(nlogn)$ time algorithm to decide whether P is walkable. There has been a considerable amount of research towards generalizing this problem on varied aspects. Heffernan [49] proposed a linear-time algorithm to solve this problem. Crass *et al.* studied a modified-version called $\infty-$searcher in an open-edge "corridor". Several researchers [66, 76, 77, 89, 12, 99] generalized *two-guard* problem in the setting of *rooms*, in which a room is a simple polygon with a designated point on its boundary called the *door*. In a more general sense, Suzuki and Yamashita [88] formulated the framework of *polygon search problems* and [90, 57, 98, 64, 97] contributed to this framework.

In all of these generalizations, polygon is supposed to be made of line segments, however, we observe that in real-life applications the boundaries are often curves instead of line segments. Therefore, we intend to generalize the concept of a simple polygon to a polygon with curves as its boundaries and solve the two-guard problem in that case. Such a polygon is called a *curvilinear polygon*. The idea of curvilinear polygons is introduced by Karavelas [58]. The curvilinear polygon defined in [58] is both piecewise locally convex and made up of convex arcs in order to admit the triangulation technique. However, in our application of generalization of two-guard problem, the curvilinear polygon is only required to be piecewise locally convex.

Definition 4. *Let $v_1, ..., v_n, n \geq 2$ be a sequence of points and let $a_1, ..., a_n$ be a set of curvilinear arcs such that a_i has the points v_i and v_{i+1} as endpoints and every a_i is a smooth Jordan arc. Assume that the arcs a_i and a_j $(i \neq j)$ intersect only if $i = j+1$ or $j = i+1$, and at that time they only intersect at v_i or v_{i+1}. Define a curvilinear polygon P to be the closed region of the plane delimited by the arcs a_i. The points v_i are called vertices of P. P is called a* piecewise locally convex polygon*, if for every point p on the boundary of P, except of $P's$ vertices, there exists a disk centered at p, say D_p, such that $P \cap D_p$ is convex.*

Figure 6.1 is an example of a piecewise locally convex polygon, while s is the vertex with exception. In the two-guard problem, we are interested in analyzing how to ensure visibility by avoidance of obstacles, so we would like to get rid of the impact of local structures that eliminate visibility. Thus, it is natural to require piecewise locally convexity on curvilinear polygons because this property can ensure visibility locally.

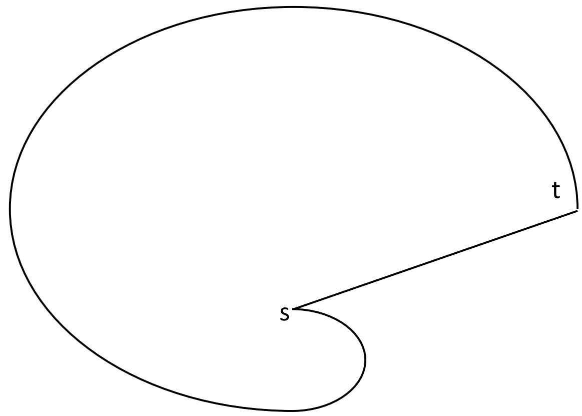


Figure 6.1: Illustration of a piecewise locally convex polygon.

The two-guard problem in piecewise locally convex polygons is stated as follows.
CURVILINEAR TWO-GUARD: Given a piecewise locally convex polygon P and s, t on the boundary of P, is P walkable from s to t?

6.2 Properties of Piecewise Locally Convex Polygons

In this section, we develop tools to solve the CURVILINEAR TWO GUARD problem.

Given a piecewise locally convex polygon P with $v_1, \ldots, v_n, n \geq 2$ as its vertices and $a_1, \ldots, a_n$ as its arcs such that a_i has the points v_i and v_{i+1} as endpoint, consider v_i and a_i (see Figure 6.2).

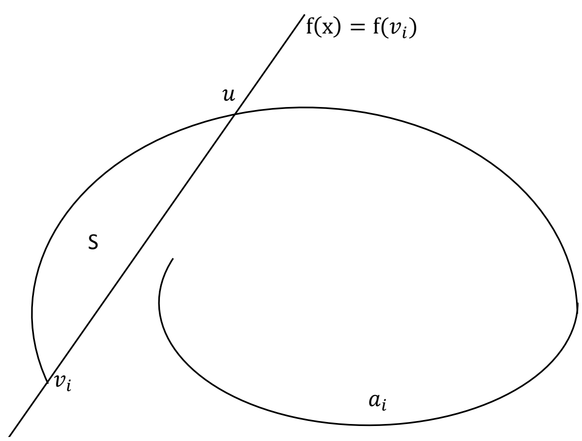


Figure 6.2: Illustration of v_i and a_i.

Proposition 14. *Suppose v_i and a_i are as above, then there exists $u \in a_i$ such that the set S made up of the line segment $\overline{v_i u}$ and the segment of a_i between v_i and u (denoted by $a_i[v_i, u]$) is a convex set.*

Proof. Because P is not necessarily locally convex in only n points $v_1, \ldots, v_n$, it is obvious that there exists $u \in a_i$ such that the open line segment $\overline{v_i u} \subset intP$ and $a_i[v_i, u]$ is entirely contained in one half-plane defined by $f(x) = f(v_i)$. Suppose W.L.O.G., for any $x \in a_i[v_i, u]$ we have $f(x) \leq f(v_i)$. Let S be the set made up of $a_i[v_i, u]$ and $\overline{v_i u}$. Then by definition, $int(S)$ is weakly supported at v_i locally with any $N(v_i)$ and f. Because S is locally convex at all points in S other than v_i, $int(S)$ is weakly supported at its boundary points other than v_i locally by Proposition 2. So by Theorem 7, $int(S)$ is convex and by Proposition 1, S is convex. □

This theorem shows $a_i[v_i, u]$ is a convex arc and by knowledge in Section 2.4, the tangent line of $a_i[v_i, u]$ at v_i exist; denote it by $T_{a_i}(v_i)$, as illustrated in Figure 6.3. By the definition of tangent lines, $T_{a_i}(v_i)$ is decided by a_i in the neighborhood of v_i, so it is only related to the choice of u. Thus we have the following definitions.

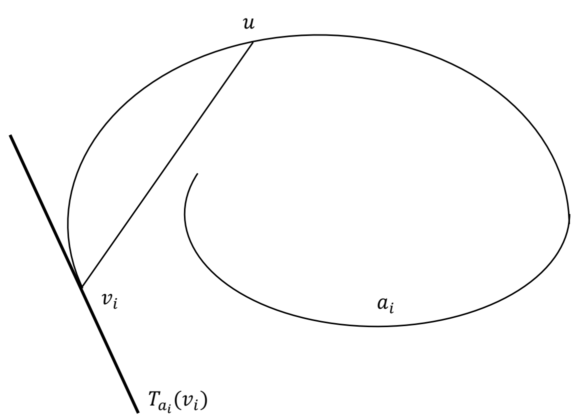


Figure 6.3: Illustration of v_i and a_i.

Definition 5. *Suppose s and t are given, two routes from s to t are called L and R.*

1. *Let v_{i-1}, v_i, v_{i+1} be three consecutive vertices on L (if $v_i = s$, v_{i-1} is the first vertex on R from s; if $v_i = t, v_{i+1}$ is the first vertex on R from t) and a_{i-1} and a_i are the arcs whose*

endpoints are v_{i-1},v_i and v_i, v_{i+1}, respectively. Let tangents at v_i as an endpoint of a_{i-1} and as an endpoint of a_i be $T_{a_{i-1}}(v_i)$ and $T_{a_i}(v_i)$, respectively.

2. *Obviously, if $T_{a_{i-1}}(v_i)$ is outside P in an area around v_i, $T_{a_i}(v_i)$ is also outside P in this area, and vice versa. At that time we call v_i a straight vertex. Otherwise we call v_i a reflex vertex.*

3. *If v_i is a reflex vertex, let $d^-(v_i)$ be the direction of $T_{a_{i-1}}(v_i)$ from outside P to inside P and let $d^+(v_i)$ be the direction of $T_{a_i}(v_i)$ from outside P to inside P. Define the first intersection point (except v_i) of the ray starting from v_i in the direction of $d^-(v_i)$ and P to be $t^-(v_i)$. Analogously, $t^+(v_i)$ is the first intersection point (except v_i) of the ray starting from v_i in the direction of $d^+(v_i)$ and P (see Figure 6.4).*

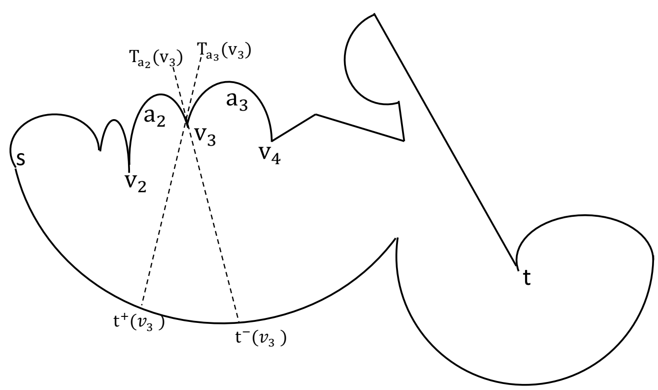


Figure 6.4: Definition of $t^+(v_i)$,$t^-(v_i)$,$T_{a_{i-1}}(v_i)$, and $T_{a_i}(v_i)$.

By Proposition 3, $T_{a_i}(v_i)$ is a supporting line of S, so S is entirely in one half-plane formed by $T_{a_i}(v_i)$. This shows a_i in a neighborhood of v_i is entirely in one half-plane formed by $T_{a_i}(v_i)$. Next lemma shows for any point u in the piecewise locally convex polygon P that lies in the different half-plane, we can find a point w on a_i in the neighborhood of v_i such that u and w are not visible. For simplicity, in next lemma we suppose a_i is entirely in one half plane formed by $T_{a_i}(v_i)$, but it is easy to see the lemma is still correct in general case.

Lemma 23.

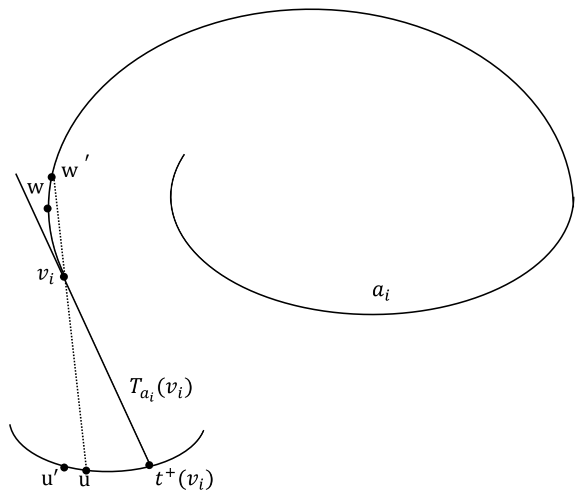


Figure 6.5: u is not visible from w.

- *Let a_i, v_i and $t^+(v_i)$ are as shown in Figure 6.5. Suppose u is on boundary of P and it is in the different half-plane formed by $T_{a_i}(v_i)$ from a_i, then there exists $w \in a_i$ such that u and w are not visible. Furthermore, if u' is also on the boundary of P and the order is u', u and $t^+(v_i)$, then w and u' are not visible.*

- *Similarly, if u is on the boundary of P and it is in the different half-plane formed by $T_{a_{i-1}}(v_i)$ from a_{i-1}, then there exists $w \in a_{i-1}$ such that u and w are not visible. Furthermore, if u' is also on the boundary of P and the order is $t^-(v_i)$, u and u', then w and u' are not visible.*

Proof. Because $T_{a_i}(v_i)$ is a tangent line of a_i, it is easy to see the line through u and v_i should interest a_i in another point; suppose it is w'. Because u is in the different half-plane formed by $T_{a_i}(v_i)$ from a_i, every line segment from u to a point on a_i in P should pass the line segment $T_{a_i}(v_i)$ from v_i to $t^+(v_i)$. Thus, it is obvious that any point $w \in a_i(v_i, w')$ is not visible from u as the line segment between u and w cannot pass the line segment $T_{a_i}(v_i)$ from v_i to $t^+(v_i)$. Furthermore, when u' is also on the boundary of P and the order is u', u and $t^+(v_i)$, if w is visible from u', also note w is visible from $t^+(v_i)$, it is easy to show that the set made of line segments $l[w, u']$, $l[w, t^+(v_i)]$ and the boundary of P from u' to $t^+(v_i)$ is convex. So, w should be visible from u, and this is a contradiction. Thus, the first statement is true.

The proof of the second statement is similar. □

6.3 Necessary Condition for P to be Walkable

In this section, we develop a necessary condition for a piecewise locally convex polygon P to be walkable. This part is quite simple and summerized in the following lemma.

Lemma 24. *Let the two chains from s to t in a piecewise locally convex polygon P be L and R. If any of the following cases happen for any reflex vertices p,q in P, then P is not walkable. (Here $p < q$ means p is before q when walking from s to t along L or R.)*

1. $p > t^-(p) \in L$ *or* $p < t^+(p) \in L$ *or* $p < t^+(p) \in R$ *or* $p > t^-(p) \in R$.

2. $p \in L\ \ q \in R\ \ q < t^+(p) \in R\ \ p < T^+(q) \in L$ *or* $p \in L\ \ q \in R\ \ q > t^-(p) \in R\ \ p > T^-(q) \in L$.

3. $p, q \in L\ \ p < q\ \ t^-(q) < t^+(p) \in R$ *or* $p, q \in R\ \ q < p\ \ t^-(p) < t^+(q) \in L$.

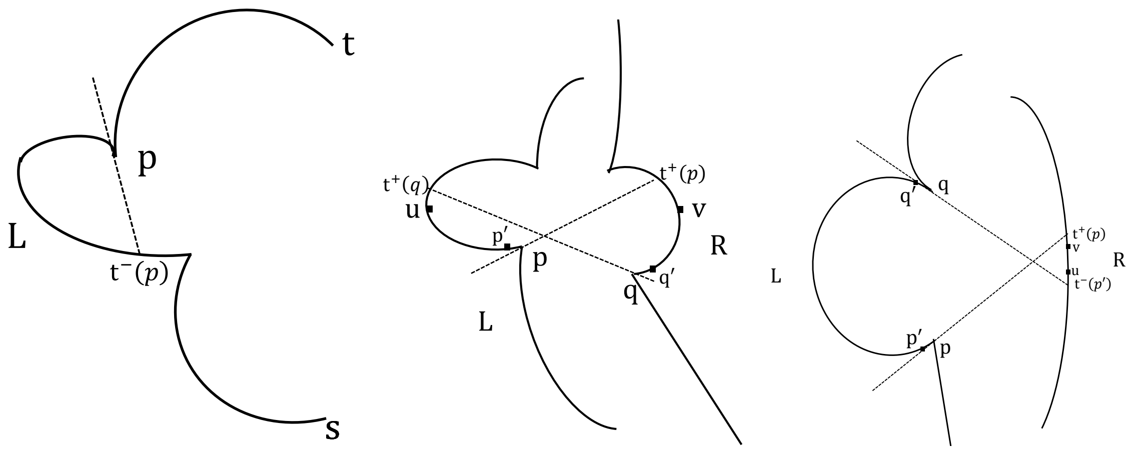


Figure 6.6: Illustration of three cases

Proof. If case 1 applies, W.L.O.G., we suppose the first alternative applies, see the left image in Figure 6.6. Denote the boundary curve through p and $t^-(p)$ by a, then by Lemma 23, for any point $p \in R$, there exists $v \in a$ such that v is invisible from p. So P is not walkable.

If case 2 applies, W.L.O.G., we suppose the first alternative applies, see the middle image in Figure 6.6. Because $q < t^+(p) \in R$, choose $v \in R$ with $q < v < t^+(p)$. By Lemma 23, there exists

$p' > p$ such that p' is not visible form v and any point in $R_{<v}$ where $R_{<v}$ means all points before v in R. So, a walking partner $\bar{p}$ of p' satisfies $\bar{p} > v$. Symmetrically, choose $u \in L$ with $p' < u < t^+(q)$, we can find q' with $q' < v$ (so $q' < \bar{p}$) whose walking partner $\bar{q}$ satisfies $\bar{q} > u > p'$. If P is walkable, $\bar{q} > p'$ means a walk go through p' before q' while $\bar{p} > q'$ means a walk go through q' before p'. This is a contradiction. So P is not walkable.

If case 3 applies, still W.L.O.G. we suppose the first alternative applies, see the right image in Figure 6.6. Choose $u, v \in R$ with $t^-(q) < u < v < t^+(p)$. As before, there exist p' whose walk partner $\bar{p} > v$ and q' whose walk partner $\bar{q} < u$. So $\bar{q} < \bar{p}$. But $p < q$, this is a contradiction. So P is not walkable.

So, if any of the three cases happens, P is not walkable. $\square$

6.4 Sufficient Conditions for P to be Walkable

In the previous section we have shown that the necessary condition for P to be walkable is that none of the cases in Lemma 24 happen for any reflex vertex. In this section, we are focused on the sufficient condition. P is assumed to be a piecewise locally convex polygon.

Definition 6. *For every reflex vertex p in L, define*

$$hiP(p) = min\{q|q \text{ is a vertex in } R, L \ni t^+(q) > p\}$$

$$hiS(p) = min\{t^-(p') \in R|p' \text{ is a vertex in } L_{>p}\}$$

$$hi(p) = min\{hiP(p), hiS(p), t\}$$

$$loP(p) = max\{q|q \text{ is a vertex in } R, L \ni t^-(q) < p\}$$

$$loS(p) = max\{t^+(p') \in R|p' \text{ is a vertex in } L_{<p}\}$$

$$lo(p) = max\{loP(p), loS(p), s\}.$$

Obviously, lo and hi are monotonically increasing functions in vertices of L.

Similarly, we can define lo and hi for vertices of R

Next two lemmata show the important relationship between lo and hi.

Lemma 25.

1. *If $q < lo(p)$ then $hi(q) < p$; if $q > hi(p)$, then $lo(q) > p$.*

2. $p \in [lo(q), hi(q)]$ *if and only if* $q \in [lo(p), hi(p)]$.

Proof. 1. First prove the first statement. If $lo(p) = loS(p)$, $q < loS(p)$. By definition of $loS(p)$, $\exists p' < p \quad q < t^+(p') \in R$, so by definition of $hiP(q)$, $hi(q) \leq hiP(q) \leq p' < p$. If $lo(p) = loP(p)$,$q < loP(p)$. So $loP(p) \in L_{>q}$ and $t^-(loP(p)) < p$, so $hi(q) \leq hiS(q) \leq t^-(loP(p)) < p$.

2. The second statement can be proved using the same methods. If $q \notin [lo(p), hi(p)]$, then $q > hi(p)$ or $q < lo(p)$. By the first statement, $p < lo(q)$ or $p > hi(q)$, and we get contradiction in both cases.

□

Lemma 26. *If none of the conditions in Lemma 24 applies in any vertex p, then* $lo(p) \leq hi(p)$ *for every vertex p in* P.

Proof. We prove by contradiction. If $lo(p) > hi(p)$ for some vertex $p \in P$, W.L.O.G., suppose $p \in L$. Then $lo(p) \neq s$ and $hi(p) \neq t$. There are four cases.

1. $hi(P) = hiP(p), \quad lo(P) = loS(P)$

   In this case, let $q = hiP(p) \in R$, then $t^+(q) > p$. $loS(p) = t^+(p')$ for some $p' \in L_p$, so $t^+(p') > q$ and $t^+(q) > p > p'$. The first alternative of condition 2 in Lemma 24 applies.

2. $hi(P) = hiS(p), \quad lo(P) = loS(P)$

   In this case, $hiS(p) = t^-(p')$ for some $p' \in L_{>p}$. $loS(p) = t^+(p'')$ for some $p'' \in L_{<p}$. So, $p'' < p'$ and $t^+(p'') > t^-(p')$. The first alternative of condition 3 in Lemma 24 applies.

3. $hi(P) = hiP(p), \quad lo(P) = loP(P)$

   In this case, $hiP(p) = q' \in R$ with $t^+(q') > p$. $loP(p) = q \in R$ with $t^-(q) < p$. So, $q' < q$ and $t^-(q) < t^+(q')$. The second alternative of condition 3 in Lemma 24 applies.

4. $hi(P) = hiS(p), \quad lo(P) = loP(P)$

   In this case, $hiS(p) = t^-(p')$ for some $p' \in L_{>p}$. $loP(p) = q \in R$ with $t^-(q') < p$. So, $t^-(p') < q$ and $t^-(q') < p < p'$. The second alternative of condition 2 in Lemma 24 applies.

So, in general, $lo(p) \leq hi(p)$ for every vertex p in P. □

Next two lemmata indicate the reason why we analyze lo and hi. In fact, these concepts play critical roles in checking whether P is walkable.

Lemma 27. *Each walk partner of a vertex p is contained in $[lo(p), hi(p)]$.*

Proof. Let $\bar{p}$ be a walk partner of p. We want to show $lo(p) \leq \bar{p} \leq hi(P)$. If $lo(p) = s$ or $hi(p) = t$, it is trivial. So, the following four situations are remaining.

1. $lo(p) = loP(p)$.

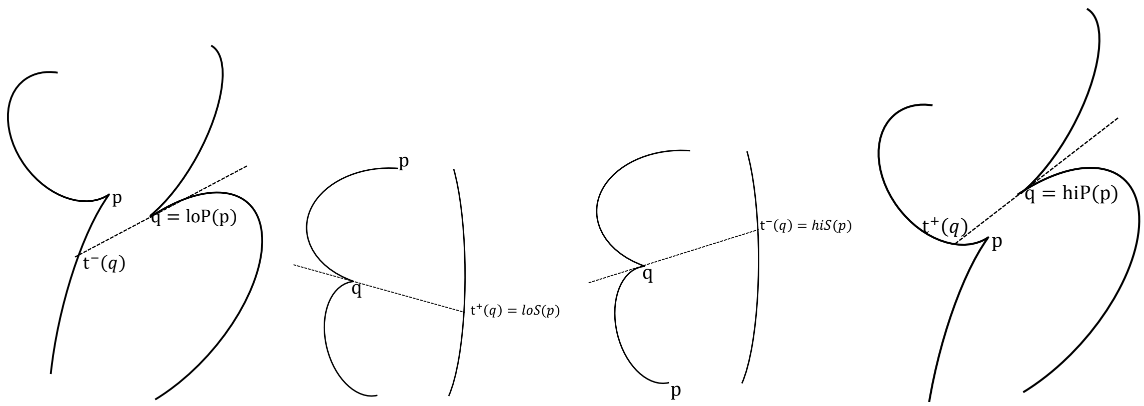


Figure 6.7: Illustration of the four situations in the proof of Lemma 27.

   See the leftmost picture of Figure 6.7. Suppose $q = loP(p)$. Then by Lemma 23, there is a sequence of points $\{q_n\}$ with $R \ni q_n \leq q$ and $q_n \to q$, q_n is not visible from $L_{\geq p}$. If $\bar{p} < q$, $\exists q_N$ s.t. $\bar{p} < q_N < q$, then q_N does not have a walk partner.

2. $lo(p) = loS(p)$.

   See the second from the left image of Figure 6.7. Suppose $t^+(q) = loS(p)$. If $\bar{p} < loS(p)$, by Lemma 23, there is a sequence of points $\{q_n\}$ with $L \ni q_n \geq q$ and $q_n \to q$, q_n is not visible from $R_{\leq \bar{p}}$. Any member of $\{q_n\}$ does not have a walk partner.

3. $hi(p) = hiS(p)$.

   See the second from the right picture of Figure 6.7. Suppose $t^-(q) = hiS(p)$. If $\bar{p} > hiS(p)$, by Lemma 23, there is a sequence of points $\{q_n\}$ with $L \ni q_n \leq q$ and $q_n \to q$, q_n is not visible from $R_{\geq \bar{p}}$. Any member of $\{q_n\}$ does not have a walk partner.

4. $hi(p) = hiP(p)$.

See right most figure of Fig 6.7. Suppose $q = hiP(p)$. Then by Lemma 23, there is a sequence of points $\{q_n\}$ with $R \ni q_n \geq q$ and $q_n \to q$,q_n is invisible from $L_{\leq p}$. If $\bar{p} > q$, $\exists q_N$ s.t. $\bar{p} > q_N > q$, then q_N does not have a walk partner.

□

Lemma 28. *Suppose condition* 1 *in Lemma 24 does not apply in any reflex vertex* $p \in P$. *If* $p \in P$ *satisfies* $lo(p) \leq hi(p)$, *then* $[lo(p), hi(p)]$ *is visible from* p.

Proof. W.L.O.G., suppose $p \in L$.

If $lo(p) = loS(p)$, $\exists L \ni p' < p$ such that $lo(p) = t^+(p')$, so p' is visible from $lo(p)$. If $lo(p) = loP(p)$, let $p' = t^-(loP(p))$, then $p' < p$ and p' is visible from $lo(p)$. If $lo(p) = s$, let $p' = s$, p' is visible from $lo(p)$. Thus, $\exists L \ni p' < p$, p' is visible from $lo(p)$. Similarly, $\exists L \ni p'' > p$, p'' is visible from $hi(p)$.

If p is not visible from $lo(p)$, the boundary of P must intersects $\overline{plo(p)}$. If $L_{>p''} \cup R_{>hi(p)}$ intersects $\overline{plo(p)}$, it must intersect $\overline{p''hi(p)}$, so p'' is not visible from $hi(p)$, which is a contradiction. If $L_{<p'} \cup R_{<lo(p)}$ intersects $\overline{plo(p)}$, it must intersect $\overline{p'lo(p)}$, so p' is not visible from $lo(p)$, which is also a contradiction.

If $L_{[p',p]}$ intersects $\overline{plo(p)}$, then there is a vertex $p''' \in L_{[p',p]}$ such that $L \ni t^+(q) > p$ or $R \in t^+(q) > lo(p)$, both are contradictions. For the same reason, $L_{[p,p'']}$ does not intersect $\overline{plo(p)}$. If $R_{[lo(p),hi(p)]}$ intersects $\overline{plo(p)}$, there is a vertex $q \in R_{[lo(p),hi(p)]}$ such that $R \ni t^-(q) < lo(p)$ or $L \in t^-(q) < p$, both are contradictions. So, none of $L \cup R$ intersects $\overline{plo(p)}$, and thus p is visible from $lo(p)$. Similarly, p is visible from $hi(p)$.

$\forall q \in [lo(p), hi(p)]$, by definition of lo and hi, we have $t^-(q) \geq p$ and $t^+(q) \leq p$. Now, if the boundary of P intersects $\overline{pq}$, it must intersect one of $\overline{plo(p)}$,$\overline{phi(p)}$,$\overline{t^-(q)q}$ and $\overline{t^+(q)q}$, but all of them cause contradictions. Therefore, $[lo(p), hi(p)]$ is visible from p. □

Now we are ready to present the sufficient condition for P to be walkable.

Lemma 29. *Let the two chains from* s *to* t *in* P *be* L *and* R. *If none of the cases in Lemma 24 applies, then* P *is walkable.*

Proof. We show P is walkable by construction of a straight walk. This task is equivalent to finding a walk instruction that decides the location of two guards at each time moment to keep them visible to each other. First, we partition P in small pieces and discuss the walk instruction for each small piece.

It is proved in Lemma 26 that $lo(p) \leq hi(p)$ for every vertex p. Then it follows by Lemma 28 that $[lo(p), hi(p)]$ is visible from p. Choose $lo(p)$ to be a walk partner of p for every vertex p in L. Because lo is monotonically increasing in L, no two walk line segments cross.

For every vertex $q \in R$, if it does not have a walk partner yet, then there exist consecutive $p, p' \in L$ with $p < p'$ and $lo(p) < q < lo(p')$. It follows from Lemma 25 that $p < hi(p) < p'$. Choose $hi(q)$ to be walk partner of q so that no pairs of $\overline{plo(p)}$ and $\overline{qhi(q)}$ will cross. Since hi is also monotonic, no two walk line segments will cross.

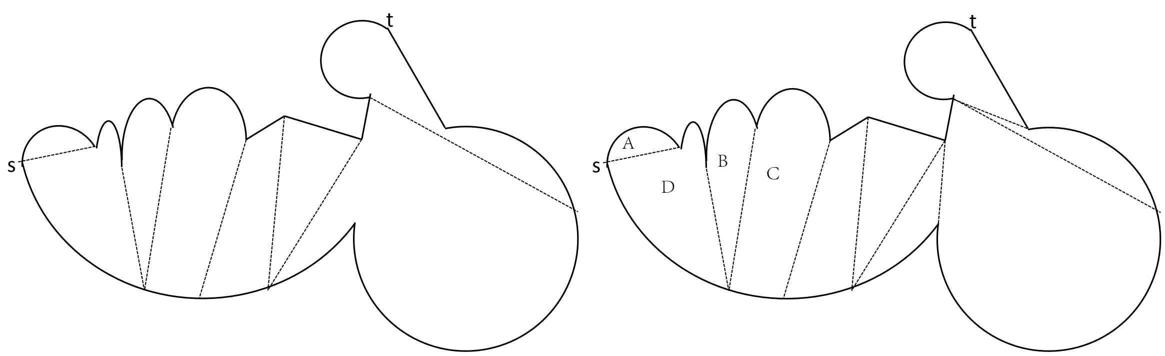


Figure 6.8: Partition when all vertices in L have partners (left) and all vertices have partners (right).

Now P is partitioned into a sequence of lenses (made of two curves), curvilinear triangles (made of three curves), and quadrilaterals (made of three curves). Figure 6.8 is an example of such a partition. Set A is an example of a lens, set B is an example of a curvilinear triangle, and sets C and D are examples of quadrilaterals.

For a lens A, it is obvious that one (non-smooth) vertex must be s or t. In Lemma 14, we have already shown A is convex, so W.L.O.G., assume s is a vertex in A and one curve of A is a part of L. The walk instruction is to keep the guard on R at S while the guard on L moves from s to the other end point.

Next we need to present a walk instruction in every curvilinear triangle and quadrilateral.

For curvilinear triangles, if one of three vertices p is in L and the other two q_1 and q_2 ($q_1 < q_2$) are in R, we need to show $q_1, q_2 \in [lo(p), hi(p)]$. There are three cases.

1. $q_1 = lo(p)$ and $q_2 = hi(p)$.

   If $hi(p) < q_2$, then by Lemma 25 $lo(q_2) > p$, which is a contradiction. So $lo(p) = q_1 < q_2 \leq hi(p)$, and the walk instruction is to keep one guard in p and let the other guard walk from q_1 to q_2.

2. $q_2 = lo(p)$ and $hi(q_1) = p$.

   It means $q_1 < lo(p)$, by Lemma 25 $hi(q_1) < p$, so this case is impossible.

3. $p = hi(q_1) = hi(q_2)$.

   If $lo(p) > q_1$,by Lemma 25 $hi(q_1) < p = hi(q_1)$, which is a contradiction. If $hi(p) < q_2$, by Lemma 25 $lo(q_2) > p = hi(q_2)$, which is also a contradiction. So, $lo(p) = q_1 < q_2 \leq hi(p)$ for the same reason as in case 1, and we can generate the walk instruction.

If two of the three vertices, p_1 and p_2 ($p_1 < p_2$) are in L and the other one q is in R, then $lo(p_1) = lo(p_2) = q$. If $p_1 < lo(q)$, by Lemma 25 $hi(p_1) < q = lo(p_1)$, which is a contradiction. If $p_2 > hi(q)$, by Lemma 25 $lo(p_2) > q = lo(p_2)$, which is also a contradiction. So, $lo(q) = p_1 < p_2 \leq hi(q)$, then the triangle is convex and it is easy to generate the walk instruction.

Each quadrilateral Q is made up of two consecutive vertices $p < p' \in L$ and two points $q < q' \in R$. Thus $q = lo(p)$ or $p = hi(q)$; $q' = lo(p')$ or $p' = hi(q')$. If Q is not locally convex in p, then p must be a reflex vertex in P. At that time, if $t^+(p) > q'$, then by definition of $loS(p')$, $lo(p') > q'$. By Lemma 25, $hi(q') > p'$. They contradict to both cases of $q' = lo(p')$ or $p' = hi(q')$. Thus $t^+(p) \leq q'$. Similarly, when Q is not locally convex in either of q, p', q', we have $t^+(q) \leq p'$, $t^-(p') \geq q$, and $t^-(q') \geq p$, respectively.

If Q is not locally convex in both p and q, then $t^+(q) > p$ and $t^+(p) > q$, and case 2 in Lemma 24 applies. So, Q must be locally convex in at least one of p, q. Similarly, Q must be locally convex in at least one of p, q. If Q is locally convex in p, q, p', and q', the quadrilateral is convex since it is locally convex in all boundary points. If Q is not locally convex in only one of p, q, p', q', say p, then $q < t^+(p) \leq q'$. The triangle made up of p, q and $t^+(p)$ and the quadrilateral made up of $p, t^+(p), p', q'$ are both convex since it is locally convex in all boundary points. If Q is not locally convex in one of p, q and one of p', q', by symmetry, there are two cases.

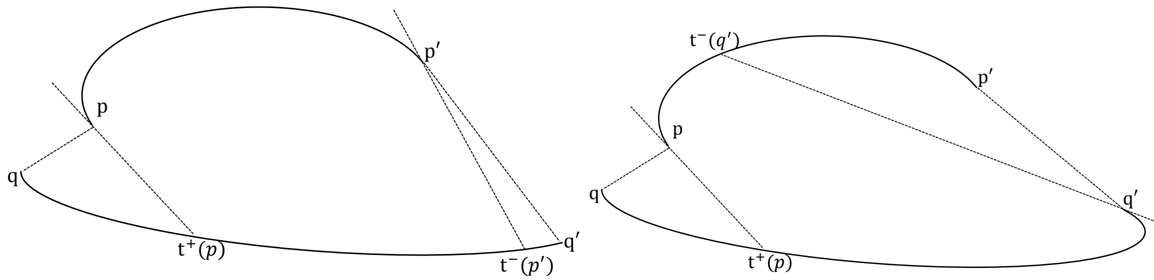


Figure 6.9: Walk instruction of quadrilaterals.

1. p, p' are the point of local non-convexity. See Figure 6.9 (left). Then $q < t^+(p), t^-(p') \le q'$. If $t^+(p) > t^-(p')$, case 3 in Lemma 24 apples, so $t^+(p) \le t^-(p')$. Then the triangle made up of p, q and $t^+(p)$, the triangle made up of p', q' and $t^-(p')$ and the quadrilateral made up of $t^-(p'), t^+(p), p', q'$ are all convex due to local convexity in all boundary points.

2. p, q' are the point of local non-convexity. See Figure 6.9 (right). At that time $q < t^+(p) \le q'$ and $p < t^+(q) \le p'$.The triangle made up of p, q and $t^+(p)$, the triangle made up of p', q' and $t^-(q')$ and the quadrilateral made up of $p, t^+(p), p', q'$ are all convex since all boundary points are locally convex.

In each case, we divide Q into at most 3 convex pieces, each of which obviously admits a walk instruction. Putting them together, we get a walk instruction for Q.

Now we generate the walk instruction for every piece, and putting the piece instructions together we get a walk instruction for P, so P is walkable. □

6.5 Construction of Solutions

In this section, we summarize the results in previous sections and develop an algorithm to check whether a piecewise locally convex polygon is walkable in quadratic time. We also develop an algorithm to generate the walk instruction if the polygon is walkable in quadratic time.

Theorem 20. *Let the two chains from s to t in a piecewise locally convex polygon P with n reflex vertices be L and R. P is walkable if and only if none the cases in Lemma 24 applies. With tangent information of reflex vertices of P at hand, there is an algorithm running in time $\mathcal{O}(n^2)$ to check whether P is walkable.*

Proof. Combining Lemma 24 and Lemma 29, we know P is walkable if and only if none the cases in Lemma 24 applies.

To check the conditions in Lemma 24, for each reflex vertex p that is the intersection of boundary curves a and b, it takes $\mathcal{O}(n)$ time to compare the intersection points of $T_a(p)$, $T_b(p)$ and every boundary curve other than a,b to derive $t^-(p)$ and $t^+(p)$ by the similar method as in [45]. So it takes $\mathcal{O}(n^2)$ time to derive $t^-(p)$ and $t^+(p)$ for all reflex vertices.

With information of $t^-(p)$ and $t^+(p)$ for every reflex vertex p, we need $\mathcal{O}(n)$ time to check condition 1 in Lemma 24 as we only need to compare p with $t^-(p)$ and $t^+(p)$ for the n reflex vertices. It takes $\mathcal{O}(n^2)$ time to check condition 2 in Lemma 24 as we need to compare each pair of p, q with $t^-(p)$,$t^+(p)$, $t^-(q)$ and $t^+(q)$. For the same reason, it takes $\mathcal{O}(n^2)$ time to check condition 3 in Lemma 24. So, the total time required to check whether P is walkable is $\mathcal{O}(n^2)$. $\square$

Corollary 5. *There is an algorithm running in time $\mathcal{O}(n^2)$ to construct a walk instruction if P is walkable.*

Proof. See Algorithm 1.

Algorithm 1 Construction of a walk instruction.

1: Derive $t^-(p)$ and $t^+(p)$ for every reflex vertex p.
2: Calculate $hi(p)$ and $lo(p)$ for every reflex vertex p.
3: For every reflex vertex $p \in L$, connect p and $lo(p)$; then if for some reflex vertex $q \in R$, q is not connected with any reflex vertex $p \in L$, connect q and $hi(q)$. As a result, P is partitioned into small pieces.
4: Construct a walk instruction for every piece.

By Theorem 20, step 1 takes $\mathcal{O}(n^2)$ time. With known $t^-(p)$ and $t^+(p)$ and by definition of lo and hi, it takes $\mathcal{O}(n^2)$ time to finish step 2. Obviously step 3 needs $\mathcal{O}(n)$ time and the resulting small pieces are lenses, curvilinear triangles and quadrilaterals by Lemma 29. The total number of small pieces is at most $2n$ and by Lemma 29, it takes $\mathcal{O}(1)$ time to construct a walk instruction for every small piece, so the total time required for step 4 is $\mathcal{O}(n)$. Therefore, the time complexity of this algorithm is $\mathcal{O}(n^2)$. The correctness of this algorithm follows directly from Lemma 29 and Theorem 20. $\square$

7. CONCLUSION AND FUTURE WORK

This dissertation explored connectivity requirements and connectivity related problems arising in network analysis. It investigated the structural properties of vertex connected subgraph polytope in a systematic fashion for the first time. Knowledge of these properties can be utilized within classical branch-and-cut techniques in mathematical programming for developing effective integer programming algorithms to solve network analysis problems. This dissertation further searched for subclasses of networks whose connected subgraph polytopes can be fully described, which simplifies the study of connectivity requirements in such subclasses significantly. In addition, this dissertation investigated connectivity related problems in different geometric network models. The unit disk graph model and visibility problems were studied. Tools that take advantage of the varied geometric structures were developed. We now go into more detail about our precise contributions and discuss areas for possible future research.

7.1 Facets of Connected Subgraph Polytope

The first contribution of this dissertation is the extensive study of facets of a vertex connected subgraph polytope $\mathscr{P}(G)$. We noted the inequality $\sum_{j \in S} x_j \leq 1$ induces a facet of $\mathscr{P}(G[S])$ while S is an independent set of G and expanded it to a facet of $\mathscr{P}(G)$ by applying a type of lifting principle sequentially. We showed this lifting procedure could generate a large class of facets of $\mathscr{P}(G)$, especially the widely-used vertex-separator inequalities $x_a + x_b - \sum_{j \in C} x_j \leq 1$, where C is a minimal a, b-separator, could be derived by this procedure. This investigation answered the open question when the vertex-separator inequalities are tight (i.e. define facets of $\mathscr{P}(G)$). Meanwhile, we saw that not all facets can be generated by the lifting procedure. In fact, we showed the right side of a facet-defining inequality is unbounded while all positive coefficients are kept at 1.

We considered the computational complexity of the lifting procedure. We proved this procedure is $\mathcal{NP}$-hard in general, which means general graphs and general lifting orders. Our result shows the procedure remains $\mathcal{NP}$-hard when the graphs are restricted to be bipartite and 2-degenerate. On the other side, we presented a linear algorithm to do lifting when graphs are acyclic graphs (i.e. 1-degenerate). Also we showed for every graph there exists a specified order such that the lifting procedure takes polytime.

There are virous directions for future research. Firstly, although not all facets can be generated by lifting, it is open whether all facets with right-hand side 1 can be generated by lifting. Our preliminary computational experiment showed about 80 percent of facets have right-hand side 1, so the power of lifting procedure will be more convincing if the conjecture is true. Secondly, it is interesting if we could also apply different types of lifting principle. The main idea of the lifting principle we apply is to fix one variable to zero at first, and then relax it. It is natural to consider another lifting principle: fix one variable to one at first, and then relax it. The new lifting principle may bring potential benefits because the right-hand side is expected to change when applying the new lifting principle, while it remains the same now, resulting in a larger class of facets. Finally, an interesting and challenging task is to generalize the study of connected subgraph polytope to k-connected subgraph polytope. The generalization is not trivial, for example, the corresponding vertex separator inequality $kx_a + kx_b - \sum_{j\in C} x_j \leq k$ is valid for k-connected subgraph polytope, but is not facet-defining.

7.2 Description of Connected Subgraph Polytope

The second contribution of this dissertation is in fully describing $\mathscr{P}(G)$ for special graphs by linear inequalities. Such a description plays an important role in integer programming because with the description, we are able to omit the binary constraints that are usually necessary in graph-related optimization problems, so that linear programming techniques can be utilized to simplify the study of connectivity requirements significantly. In this dissertation, we discussed full description for three classes of graphs: the first one is graphs whose independence number $\alpha(G) \leq 2$; the second one is *3-plex* and *3-defective cliques*; the third one is acyclic graphs.

When $\alpha(G) \leq 2$, vertex separator inequalities together with bound inequalities characterize $\mathscr{P}(G)$. If we define

$$Q(G) := \left\{ x \in [0,1]^{|V|} \;\middle|\; x \text{ satisfies all vertex separator inequalities} \right\},$$

then $\mathscr{P}(G) = Q(G)$. We were also interested in the structure of $Q(G)$ for general G. The number of linear inequalities to describe $Q(G)$ may be exponential with respect to $|V(G)|$ (although linear optimization over $Q(G)$ is polytime solvable by ellipsoid method), however, we were able to derive an extended formulation of $Q(G)$ whose number of linear inequalities is $\mathcal{O}(|V(G)|^4)$ by utilizing knowledge of network flow theory.

If G is a complete graph, $\alpha(G) = 1$, so G with $\alpha(G) \leq 2$ is a type of clique relaxation. Thus it is natural to consider other clique relaxations. As a result, we explored the full description of $\mathscr{P}(G)$ for a 3-plex and a 3-defective clique. In those two cases, only one kind of extra inequalities is necessary to characterize $\mathscr{P}(G)$ besides vertex separator inequalities and bound inequalities.

When G is an acyclic graph, we developed one concise necessary and sufficient condition for an inequality to define a facet of $\mathscr{P}(G)$. This condition could be converted into a linear time algorithm to generate every facet of $\mathscr{P}(G)$. In addition, we derived closed-form characterization of $\mathscr{P}(G)$ for a path graph and a star graph G.

In the future, the relationship between $\mathscr{P}(G)$ and $Q(G)$ is of interest to us. $Q(G)$ is a linear relaxation of $\mathscr{P}(G)$, so an interesting question is how good is this relaxation. It is obvious $\mathscr{P}(G) \subseteq Q(G) \subseteq c\mathscr{P}(G)$ when $c = |V(g)|$, and we hope for $c = o(|V(G)|)$. We also conjecture c is a function of $\alpha(G)$ because for $\alpha(G) = 2$, we have proved $c = 1$ and for $\alpha(G) = |V(G)|$, $Q(G) \nsubseteq c\mathscr{P}(G)$ for any $c < |V(G)|$. Meanwhile, it would also be interesting if we could generalize our characterization of $\mathscr{P}(G)$ to G with $\alpha(G) = 3, 4, \ldots$. We think it is possible because $MWCS$ is polytime solvable when $\alpha(G)$ is bounded [94].

7.3 2-Cliques In Unit Disk Graphs

The third contribution of this dissertation is in establishing a highly effective approximation algorithm for solving the maximum 2-clique problem on unit disk graphs, which are often used to model wireless communication networks. Our algorithm is proved to have a $\frac{1}{2}$ approximation ratio in the worst case, however, it appears is more effective in practice. In fact, the algorithm found the exact solution for all of the numerous random unit disk graphs we used in experiments.

There are many interesting problems concerning $2-clique$ and, more generally, $k-$clique problem on unit disk graphs left for future research. One immediate question which still needs to be answered is, what is the computational complexity of the maximum $k-$clique problem in unit disk graphs? This question is still open even for $k = 2$. Interestingly, the notoriously difficult in general maxim clique problem (i.e., $k = 1$) can be solved in polynomial time on unit disk graphs. We conjecture that the maximum $k-$clique problem is $\mathcal{NP}$-hard on unit disk graphs for $k \geq 2$. However, establishing this appears to be very challenging.

Another question to investigate is whether a better analysis of approximation ratio can be done to improve the factor from $\frac{1}{2}$ to $\frac{2}{3}$ by showing that a 2-clique in unit disk graphs is, in fact, 3-dominated. So far, we were able to establish 4-domination only, however, we failed to produce an example where more than 3 dominating nodes are required.

Developing algorithms for closely related maximum 2-club problem is another interesting topic for future research.

7.4 The Two-Guard Problem in Curvilinear Polygons

The forth contribution of this dissertation is in generalizing the two-guard problem to piecewise locally convex polygons. The two-guard problem is a visibility problem which asks whether a polygon is walkable. This problem, like other visibility problems, are usually solved for simple polygons. We generalized the concept of a simple polygon to piecewise locally convex polygon. By carefully analyzing the properties of piecewise locally convex polygons, we were able to develop tools necessary to solve the two-guard problem on such curvilinear polygons. We presented an algorithm running in quadratic time to decide whether a piecewise locally convex polygon is walkable. In addition, we derived another algorithm running in quadratic time that generates a valid walk if the polygon is walkable.

There exists an algorithm running in time $\mathcal{O}(nlogn)$ to solve the original two-guard problem but it cannot be generalized to solve our problem. Instead, our algorithm runs in quadratic time. It is an interesting topic for future research if the running time of our algorithm can be improved from quadratic time to $\mathcal{O}(nlogn)$ time. Such an improvement requires improvement on shortest path queries in a curvilinear polygon, which is itself an interesting problem in computational geometry.

There are many modified versions and generalizations of the two-guard problem, and all of them assume that the polygon is simple, defined by line segments. As our generalization considers curvilinear polygons, it is natural to consider curvilinear polygons in the modified or generalized two-guard problems in the future research. These include the two-guard problem in counter-walk polygons, the two-guard problem in the setting of rooms, and polygon search problems.

REFERENCES

[1] R.K. Ahuja, T.L. Magnanti, and J.B. Orlin. *Network flows: Theory, Algorithms and Applications*. Prentice Hall, 1993.

[2] E. Álvarez-Miranda, I. Ljubić, and P. Mutzel. The maximum weight connected subgraph problem. In M. Jünger and G. Reinelt, editors, *Facets of Combinatorial Optimization*, pages 245–270. Springer, 2013.

[3] M. Baïou, F. Barahona, and A.R. Mahjoub. Partition inequalities:separation, extensions and network design. *Prog.Comb.Optim.*, pages 1–39, 2011.

[4] M. Baïou and A.R. Mahjoub. Steiner 2 edge connected subgraph polytopes on series-parallel graphs. *SIAM J. Discrete Mathematics*, 10:505–514, 1997.

[5] E. Balas. An additive algorithm for solving linear programs with zero-one variables. *Operation Research*, 6:517–546, 1965.

[6] B. Balasundaram and S. Butenko. Optimization problems in unit-disk graphs. In C. A. Floudas and P. M. Pardalos, editors, *Encyclopedia of Optimization*, pages 2832–2844. Springer, 2^{nd} edition, 2009.

[7] B. Balasundaram, S. Butenko, and I.V. Hicks. Clique relaxations in social network analysis: The maximum k-plex problem. *Operations Research*, 59:133–142, 2011.

[8] B. Balasundaram, S. Butenko, and S. Trukhanov. Novel approaches for analyzing biological networks. *Journal of Combinatorial Optimization*, 10(1):23–39, 2005.

[9] F. Barahona and A.R. Mahjoub. On two-connected subgraph polytopes. *Discrete Mathematics*, 147:19–34, 1995.

[10] J. Bentley. Programming pearls: algorithm design techniques. *Communications of the ACM*, 27(9):865–873, 1984.

[11] D. Bertsekas. *Convex Analysis and Optimization*. Athena Scientific, 2003.

[12] B. Bhattacharya, J. Zhang, Q. Shi, and T. Kameda. An optimal solution to room search problem. In *18th Canadian Conf. on Computational Geometry,*, pages 55–58, 2006.

[13] M.D. Biha, L.M. Kerivin, and P.H. Ng. Polyhedral study of the connected subgraph problem. *Discrete Mathematics*, 338:80–92, 2015.

[14] M.D. Biha and A.R. Mahjoub. The k-edge connected subgraph problem i: Polytopes and critical extreme points. *Linear Algebra and its Applications*, 381:117–139, 2004.

[15] A. Bondy and U.S.R. Murty. *Graph Theory*. Springer, 2008.

[16] S. Boyd and T. Hao. An integer polytope related to the design of survivable communication networks. *SIAM J. Discr. Math.*, 6:612–630, 1993.

[17] H. Breu. *Algorithmic Aspects of Constrained Unit Disk Graphs*. PhD thesis, University of British Columbia, 1996.

[18] G.S. Brodal and A.G. Jørgensen. A linear time algorithm for the k maximal sums problem. In *Mathematical Foundations of Computer Science 2007*, pages 442–453. Springer, 2007.

[19] A. Buchanan. *Parameterized Approaches for Large-scale Optimization Problems*. PhD thesis, Texas A&M University, 2015.

[20] A. Buchanan, Y. Wang, and S. Butenko. Exact algorithms for node-weighted steiner tree and maximum-weight connected subgraph. Manuscript, 2014.

[21] R. Carvajal, M. Constantino, M. Goycoolea, J.P. Vielma, and A. Weintraub. Imposing connectivity constraints in forest planning models. *Operations Research*, 61(4):824–836, 2013.

[22] S. Ceroi. Clique number and ball containment number of unit ball graphs. *Electronic Notes Discrete Math*, pages 22–25, 2001.

[23] C.Y. Chen and K. Grauman. Efficient activity detection with max-subgraph search. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1274–1281, Providence, RI, USA, June 2012. IEEE.

[24] W. Chen, M.L. Giger, and U. Bick. A fuzzy c-means (FCM)-based approach for computerized segmentation of breast lesions in dynamic contrast-enhanced mr images. *Academic radiology*, 13:63–72, 2006.

[25] S. Chopra. The k-edge connected spanning subgraph polyhedron. *SIAM J. Discrete Mathematics*, 7:245–259, 1994.

[26] B.N. Clark, C.J. Colbourn, and D.S. Johnson. Unit disk graphs. *Discrete Math.*, 86:165–177, 1990.

[27] W. Cook, W. Cunningham, W. Pulleyblank, and A. Schrijver. *Combinatorial Optimization.* John Wiley and Sons, New York, 1998.

[28] L. Danzer, B. Grunbaum, and V. Klee. Helly's theorem and its relatives. *Convexity*, pages 101–180, 1963.

[29] R. Diestel. *Graph theory.* Springer-Verlag, Berlin, 1997.

[30] M.T. Dittrich, G.W. Klau, A. Rosenwald, T. Dandekar, and T. Müller. Identifying functional modules in protein–protein interaction networks: an integrated exact approach. *Bioinformatics*, 24(13):i223–i231, 2008.

[31] R. Euler, M. Jünger, and G. Reinelt. Generalizations of cliques, odd cycles and anticycles and theire relations to indepedent system polyhedra. *Mathematics of Operation Research*, 12:451–462, 1987.

[32] J. Feigenbaum, C. Papadimitriou, and S. Shenker. Sharing the cost of multicast tansmissions. *J. Comput. System Sci.*, 63:21–41, 2001.

[33] L.R. Ford and D.R. Fulkerson. *Flows in Networks.* Princeton University Press, New Jersey, 1962.

[34] A. Frank. Connectivity and network flows. *Handbook of Combinatorics*, 1:111–177, 1995.

[35] M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W.H. Freeman and Company, 1979.

[36] R.S. Garfinkel and G.L. Nemhauser. Optimal political districting by implicit enumeration techniques. *Management Science*, 16(8):B–495, 1970.

[37] A. Ghosh and S.K. Das. Coverage and connectivity issues in wireless sensor networks: A survey. *Pervasive and Mobile Computing*, 4:303–334, 2008.

[38] S.K. Ghosh. *Visibility Algorithms in the Plane.* Cambridge University Press, 2007.

[39] M. Goemans and D. Williamson. A general approximation technique for constrained forest problems. *SIAM J. Comput.*, 24:296–317, 1995.

[40] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization, 2nd edn.* Springer-Verlag, Berlin, 1993.

[41] M. Grötschel, C.L. Monma, and M. Stoer. Computational results with a cutting plane algorithm for designing communication networks with low-connectivity constraints. *Operation Research*, 40:309–330, 1992.

[42] M. Grötschel, C.L. Monma, and M. Stoer. *Design of Suvivable Networks.* Springer, 1992.

[43] M. Grötschel, C.L. Monma, and M. Stoer. Facets for polyhedra arising in the design of communication networks with low-connectivity requirements. *SIAM J. Optim.*, 2:474–504, 1992.

[44] M. Grötschel, C.L. Monma, and M. Stoer. Polydedral and computational investigations for designing communication networks with high survivability requirements. *Operation Research*, 43:1012–1024, 1992.

[45] L.J. Guibas and J. Hershberger. Optimal shortest path queries in a simple polygon. *Journal of Computer and Systems Sciences*, 39(2):126–152, 1989.

[46] H. Hadwiger, H. Debrunner, and V. Klee. *Combinatorial Geometry in the Plane.* Holt, Rinehart and Winston, 1964.

[47] S. Handy, K. Butler, and R.G. Paterson. Planning for street connectivity-getting form here to there. In *American Planning Association*, Chicago, 2003.

[48] J. Hansen, E. Schmutz, and L. Sheng. Covering random points in a unit disk. *Adv. Appl. Probab.*, 40:22–30, 2008.

[49] P. Heffernan. An optimal algorithm for the two-guard problem. *International Journal of Computational Geometry & Applications*, 6:15–44, 1996.

[50] P.M. Hess. Measures of connectivity. *Places*, 11:58–65, 1997.

[51] J.E. Hopcroft and R.M. Karp. An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. *SIAM Journal on Computing*, 2:225–231, 1973.

[52] F.K. Hwang, D.S. Richards, and P. Winter. *The Steiner Tree Problem.* Elsevier Science Publisher, 1992.

[53] C. Icking and R. Klein. The two guards problem. *International Journal of Computational Geometry & Applications*, 2:257–285, 1992.

[54] D.S. Johnson. The NP-completeness column: an ongoing guide. *Journal of Algorithms*, 6(1):145–159, 1985.

[55] D.S. Johnson, M. Minkoff, and S. Phillips. The prize collecting steiner tree problem: theory and practice. In *11th Ann. ACM-SIAM symp. on Discrete Algorithms*, pages 760–769, 2000.

[56] R. Johnson. A circle theorem. *American Mathematical Monthly*, 23:161–162, 1916.

[57] T. Kameda, J. Zhang, and M. Yamashita. Simple characterization of polygons searchable by 1-searcher. In *18th Canadian Conf. on Computational Geometry*, pages 113–116, 2006.

[58] M.I. Karavelas. Guarding curvilinear art galleries with edge or mobile guards via 2-dominance of triangulation graphs. *Computational Geometry*, 44(1):20–51, 2011.

[59] H. Kerivin and S.K. Das. Design of survivable networks: A survey. *Networks*, 46(1):1–21, 2005.

[60] H.L.M. Kerivin and P.H. Ng. Maximum-weight connected subgraph problems. Technical report, Univerisity Blaise Pascal, Clermont-Ferrand, France, 2013.

[61] L.G. Khachian. A polynomial algorithm in linear programming. *Soviet Mathematics Doklady*, 20:1093–1096, 1979.

[62] D. Kim, Y. Wu, Y. Li, F. Zou, and D.Z. Du. Constructing minimum connected dominating sets with bounded diameters in wireless networks. *IEEE Transactions on Parallel and Distributed Systems*, 20(2):147–157, 2009.

[63] M. Laurent. A genealization of antiwebs to independence systems and their canonical facets. *Mathematical Programming*, 45:97–108, 1989.

[64] S.M. LaValle, B. Simov, and G. Slutzki. An algorithm for searching a polygonal region with a flashlight. *International Journal of Computational Geometry & Applications*, 12:87–113, 2002.

[65] S.R. Lay. *Convex Sets and Their Applications.* John Wiley&Sons, 1982.

[66] J. Lee, S.M. Park, and K.Y. Chwa. Searching a polygonal room with one door by a 1-searcher. *International Journal of Computational Geometry & Applications*, 10:201–220, 2000.

[67] G. MacGillivray and K. Seyffarth. Domination numbers of planar graphs. *Journal of Graph Theory*, 22:213–229, 1996.

[68] D. Mackenzie. Triquetras and porisms. *College Math.J.*, 23:118–131, 1992.

[69] A.R. Mahjoub. Two-edge connected spanning subgraphs and polyhedra. *Mathematical Programming*, 64:199–208, 1994.

[70] A. Moilanen and M. Nieminen. Simple connectivity measures in spatial ecology. *Ecology*, 83:1131–1145, 2002.

[71] G.L. Nemhauser and L.E. Trotter. Properties of vertex packings and independence system. *Mathematical Programming*, 6:48–61, 1974.

[72] G.L. Nemhauser and L.A. Wolsey. *Integer and Combinatorial Optimization*. Wiley, 1988.

[73] P.R.J. Östergård. A fast algorithm for the maximum clique problem. *Discrete Applied Mathematics*, 120:197–207, 2002.

[74] M.W. Padberg. On the facial structure of set packing polyhedra. *Mathematical Programming*, 5:199–215, 1973.

[75] H.C. Papadimitriou and K. Steigliz. *Combinatorial optimization: algorithms and complexity*. Courier Dover Publications, 1998.

[76] S.M. Park, J. Lee, and K.Y. Chwa. Characterization of rooms searchable by two guards. In *Int. Symp. on Algorithms and Computation*, pages 515–526, 2000.

[77] S.M. Park, J. Lee, and K.Y. Chwa. Searching a room by two guards. *International Journal of Computational Geometry & Applications*, 12:339–352, 2002.

[78] J. Pattillo. *Mathematical Foundations and Algorithms for Clique Relaxations in Networks*. PhD thesis, Texas A&M University, 2011.

[79] J. Pattillo, A. Veremyev, S. Butenko, and V. Boginski. On the maximum quasi-clique problem. *Discrete Applied Mathematics*, 161:244–257, 2013.

[80] J. Pattillo, Y. Wang, and S. Butenko. Approximating 2-cliques in unit disk graphs. *Discrete Applied Mathematics*, 166:178–187, 2014.

[81] J. Pattillo, N. Youssef, and S. Butenko. Clique relaxation models in social network analysis. *Handbook of Optimization in Complex Networks*, pages 143–162, 2012.

[82] J. Pattillo, N. Youssef, and S. Butenko. On clique relaxation models in network analysis. *European Journal of Operational Research*, 226:9–18, 2013.

[83] J. Philip. The probability distribution of the distance between two random points in a box. *TRITA MAT*, 7, 2007.

[84] V. Raghavan and J. Spinrad. Robust algorithms for restricted domains. *Journal of Algorithms*, 48(1):160–172, 2003.

[85] T.A. Randall and B.W. Baetz. Evaluating pedestrian connectivity for suburan sustainability. *Joural of Urban Planning and Development*, 127:1–15, 2001.

[86] H. Samet and M. Tamminen. Efficient component labeling of images of arbitrary dimension represented by linear bintrees. *IEEE Trans. Pattern Anal. Mach. Intell.*, 10:579–586, 1988.

[87] A. Schrijver. *Theory of Linear and Integer Programming.* Wiley, New York, 1986.

[88] I. Suzuki and M. Yamashita. Searching for a mobile intruder in a polygonal region. *SIAM J. on Computing*, 21:863–888, 1992.

[89] X. Tan. *Efficient algorithms for searching a polygonal room with a door*, volume 2098, chapter JCDCG2000, pages 339–350. Springer, 2001.

[90] X. Tan. *A Characterization of Polygonal Regions Searchable from the Boundary*, volume 3330, chapter IJCCGGT2003, pages 200–215. Springer, 2005.

[91] V.A. Toponogov. *Differential Geomertry of Curves and Surfaces–A Concise Guide.* Birkhäuser Mathematics, 2006.

[92] A. Veremyev and V. Boginski. Identifying large robust network clusters via new compact formulations of maximum k-club problems. *European Journal of Operational Research*, 218:316–326, 2012.

[93] A. Vergis. Manuscript, 1983.

[94] Y. Wang, A. Buchanan, and S. Butenko. On imposing connectivity constraints in integer programming. Manuscript, 2015.

[95] K. Wu, W. Koegler, J. Chen, and A. Shoshani. Using bitmap index for interactive exploration of large datasets. In *SSDBM*, pages 65–74, 2003.

[96] H. Yu, A. Paccanaro, V. Trifonov, and M. Gerstein. Predicting interactions in protein networks by completing defective cliques. *Bioinformatics*, 22:823–829, 2006.

[97] J. Zhang. The two-guard polygon walk problem. In *TAMC2009*, pages 450–459, 2009.

[98] J. Zhang and B. Burnett. Yet another simple characterization of searchable polygons by 1-searcher. In *IEEE Intl. Conf. on Robotics and Biomimetics*, pages 1244–1249, 2006.

[99] J. Zhang and T. Kameda. *A linear-time algorithm for finding all door locations that make a room searchable(extended abstract)*, volume 4978, chapter TAMC2008, pages 502–513. Springer, 2008.