Approximation Algorithms for Facility Location Problems in Graphs

A thesis submitted in partial fulfillment of
the requirements for the degree of
Doctor of Philosophy

by

Sangram Kishor Jena

to the

Department of Mathematics
Indian Institute of Technology Guwahati, India
Guwahati-781 039, Assam, India

April 2021

Approximation Algorithms for Facility Location Problems in Graphs

A thesis submitted in partial fulfillment of
the requirements for the degree of
Doctor of Philosophy

by

Sangram Kishor Jena
(Roll No. 156123017)

Under the Supervision of
Dr. Gautam Kumar Das

to the

Department of Mathematics
Indian Institute of Technology Guwahati, India
Guwahati-781 039, Assam, India

April 2021

DECLARATION

It is certified that the work contained in the thesis entitled **"Approximation Algorithms for Facility Location Problems in Graphs"** has been done by me, a student in the Department of Mathematics, Indian Institute of Technology Guwahati, India, under the guidance of **Dr. Gautam Kumar Das** for the award of Doctor of Philosophy and that this work has not been submitted elsewhere for a degree.

Place : IIT Guwahati

Date : April 21, 2021

Sangram Kishor Jena
Indian Institute of Technology Guwahati, India
Guwahati-781 039, Assam, India

CERTIFICATE

This is to certify that this thesis entitled "**Approximation Algorithms for Facility Location Problems in Graphs**" being submitted by Mr. Sangram Kishor Jena to the Department of Mathematics, Indian Institute of Technology Guwahati, India, is a record of bona fide research work under my supervision and is worthy of consideration for the award of the degree of Doctor of Philosophy of the Institute.

The results contained in this thesis have not been submitted in part or full to any other university or institute for the award of any degree or diploma.

Place : IIT Guwahati
Date : April 21, 2021

Dr. Gautam Kumar Das
Associate Professor
Indian Institute of Technology Guwahati, India
Guwahati-781 039, Assam, India

Acknowledgements

Undertaking this PhD has been a truly life-changing experience for me and it would not have been possible to reach the destination without the support and guidance that I received from many people. I would like to take this opportunity to express my deepest gratitude for their generous support.

First and foremost, I want to express my deepest appreciation and heartfelt thanks to my thesis supervisor Dr. Gautam K. Das for allowing me to carry out the research work under his supervision and introducing me to an excellent and interesting field of approximation algorithms. His guidance, continuous support, and encouragement helped me in all the time of research and writing of this thesis. I am grateful and thank him for all the guidance he provided me, both in academic and personal contexts. I could not have imagined having a better advisor for my PhD.

Besides my advisor, I would like to thank the rest of my doctoral committee: Prof. K. V. Krishna, Dr. Sushanta Karmakar, and Dr. Aryabartta Sahu for their insightful comments and encouragement, but also for the hard question which incented me to widen my research from various perspectives.

I would also like to thank Prof. Subhas C. Nandy for his collaborative work with us and being the co-author of a publication from this thesis. I enjoyed a lot discussing problems and algorithms with my co-author Dr. Ramesh K. Jallu. He is not only my senior in PhD but also like a brother to me. I thank him from the bottom of my heart for always being supportive and there for me whenever I need.

My sincere thanks to the Ministry of Human Resource Development, Government of India, for providing me financial support for pursuing PhD at IIT Guwahati. I sincerely acknowledge IIT Guwahati for providing me the necessary facilities for conducting my research work smoothly. I also thank the Head of the Department of Mathematics, IIT Guwahati, other teaching and non-teaching staff members of the Department for their support in administrative and technical works.

I would like to thank my friends, fellow research scholars, lab mates, cubical mates, etc. at IIT Guwahati for their company during my stay in the IIT campus. I thank

Pawan, Rupak, Abhijeet, Nilanjan, to name a few, with whom I spend most of my joyful time at IIT Guwahati. I acknowledge my seniors, Manjanna, Ramesh Anna, Dibakar Da, Debashish, and juniors, Sasmita and Souvik for their valuable suggestions. I am grateful to all others who have helped me in various ways directly or indirectly while doing this thesis work at IIT Guwahati.

Many thanks to my MTech friends Anil, Rajesh, and Saumya, to name a few, who constantly motivated me to undergo for a PhD.

I express my gratitude to Prof. Sukanta Pati, Prof. S. V. Rao, and Dr. Inkulu, some of the dedicated and fantastic faculties of IIT Guwahati. In the early days of my PhD, I attended lectures during my course work and enrich some of their teaching skills.

I would like to acknowledge all the teachers who help me to learn since my childhood, I would not have been here without their guidance, blessing, and support.

Last but not the least, I would like to thank my family: my parents, for all their effort they devoted to getting me an education. Without them, I may never have gotten to where I am today. I would like to thank my brother, sister, sister-in-law for their support, encouragement, and love. I also want to thank my in-laws for their unconditional support. In the end, I would like to express appreciation to my beloved wife Swayanprajna who has taken care of me all the time, shared in my stress and joy, and given me support every day. I thank her for being understanding and cooperative during times when I was not able to spend quality time with the family.

Sangram Kishor Jena

Abstract

Dominating set and its variants have been studied extensively in the literature and are of broad and current research interest to many researchers due to its wide range of applications, including, but not limited to, networks, VLSI, clustering, map labeling, coding theory, etc. In this thesis, minimum dominating set problem and some of its variants, such as maximum distance-d independent set, minimum distance-d dominating set, minimum d-distance m-tuple (ℓ, r)-dominating set, minimum vertex-edge dominating set, and minimum total dominating set problems are studied.

All the aforementioned problems are NP-hard and none of them admit constant factor approximation algorithms for general graphs, unless P = NP. This motivated us to study the problems for special graph classes. We are the first to define the d-distance m-tuple (ℓ, r)-dominating set problem in general graphs, which is a collection of problems depending on the values of d, m, ℓ, and r and all the other four problems are studied in unit disk graphs (UDGs) with known disk position for designing simple constant factor approximation algorithms with low time complexity and polynomial-time approximation schemes that outperform the existing results in the literature.

Firstly, we study the geometric maximum distance-d independent set (GMDdIS) problem, for an integer $d \geq 3$. We define the GMDdIS problem as follows:

> *Given a simple unit disk graph $G = (V, E)$ corresponding to a point set $P = \{p_1, p_2, \ldots, p_n\}$ for disk centers in the plane, find a maximum cardinality subset $I \subseteq V$ such that for every pair of vertices $p_i, p_j \in I$ the length (number of edges) of the shortest path between p_i and p_j in G is at least d.*

We show that the decision version of the GMDdIS problem (for $d \geq 3$) is NP-complete in unit disk graphs. Next, we propose a simple 4-factor approximation algorithm for GMDdIS problem in $d^2 n^{O(d)}$ time. Finally, we propose a PTAS for this problem, which produces a DdIS of size at least $\frac{1}{(1+\frac{1}{k})^2}|OPT|$ in $k^2 n^{O(k)}$ time, where $|OPT|$ is the maximum cardinality of a GMDdIS.

Secondly, we study the geometric minimum distance-d dominating set (GMDdDS) problem in unit disk graphs. We define GMDdDS as follows:

> *Given a simple unit disk graph $G = (V, E)$ corresponding to a point set $P = \{p_1, p_2, \ldots, p_n\}$ for disk centers in the plane, find a minimum cardinality subset $D \subseteq V$ such that for each vertex $p_i \in V$, there must exist a vertex $p_j \in D$, such that the length (number of edges) of the shortest path between p_i and p_j in G is at most d.*

We show that the decision version of the GMDdDS problem (for $d \geq 2$) is NP-complete in unit disk graphs. We propose a simple 4-factor approximation algorithm for GMDdDS problem in $d^2 n^{O(d)}$ time, and a PTAS for this problem, which produces a DdDS of size at most $(1 + \frac{1}{k})^2 \times |OPT|$ in $k^2 n^{O(k)}$ time, where OPT is the minimum cardinality of a GMDdDS.

Next, we study the minimum d-distance m-tuple (ℓ, r)-dominating set ((d, m, ℓ, r) set) problem in general graphs. We define (d, m, ℓ, r) set problem as follows:

> *Given a simple graph $G = (V, E)$ and positive integers d, m, ℓ and r, find a minimum size subset $V' \subseteq V$ of G satisfying the following two conditions: (i) each vertex $v \in V$ is d-distance dominated by at least m vertices in V', and (ii) each r size subset U of V is d-distance dominated by at least ℓ vertices in V'.*

Here, a vertex v is d-distance dominated by another vertex u means the shortest path distance between u and v is at most d in G. A set U is d-distance dominated by a set of ℓ vertices means the size of the union of the d-distance neighborhood of all vertices of U in V' is at least ℓ. In this problem, we prove that the problem of deciding whether a graph G has a 1-distance m-tuple (ℓ, r)-dominating set for each fixed value of m, ℓ, and r of cardinality at most k is NP-complete for general graphs. Next, we prove that the problem of deciding whether a graph G has a d-distance m-tuple $(\ell, 2)$-dominating set for each fixed value of $d(> 1), m$, and ℓ of cardinality at most k is NP-complete. We also prove that for any $\varepsilon > 0$, the 1-distance m-tuple (ℓ, r)-domination problem and the d-distance m-tuple $(\ell, 2)$-domination problem cannot be approximated within a factor of $(\frac{1}{2} - \varepsilon) \ln |V|$ and $(\frac{1}{4} - \varepsilon) \ln |V|$, respectively, unless $P = NP$.

Next, we introduce a variant of dominating set problem in UDGs and we call this problem as the geometric minimum vertex-edge dominating set (GMVEDS) problem. We define the GMVEDS as follows:

> *Given a simple unit disk graph $G = (V, E)$ corresponding to a point set $P = \{p_1, p_2, \ldots, p_n\}$ for disk centers in the plane, find a minimum cardinality subset $D \subseteq V$ such that for each edge $e = p_ip_j \in E$, either p_i or p_j is in D or one vertex from their neighbor is in D.*

Simply, a vertex $p_i \in V$, vertex-edge dominates every edge p_ip_j, as well as every edge adjacent to these edges. We prove that the problem GMVEDS belongs to the NP-hard class in unit disk graphs. We propose a simple 4-factor approximation algorithm in polynomial time. We also designe a PTAS for this problem, which runs in $O(n^c)$ time, where $c = O(\frac{1}{\epsilon} \log \frac{1}{\epsilon})$.

Finally, we study the geometric minimum total dominating set (GMTDS) problem in UDGs. We define GMTDS as follows:

> *Given a simple unit disk graph $G = (V, E)$ corresponding to a point set $P = \{p_1, p_2, \ldots, p_n\}$ for disk centers in the plane, find a minimum cardinality subset $D_t \subseteq V$ such that for each vertex $p_i \in V$, there exist a dominator $p_j \in D_t$ such that $p_i \neq p_j$ and Euclidean distance between p_i and p_j is at most one.*

In the GMTDS problem, we prove that the decision version of the TDS problem in the unit disk graph is NP-complete. Next, we propose a simple 8-factor approximation algorithm in $O(n \log k)$ time, where n is the input size and k is the output size of the algorithm. We also propose a PTAS for this problem, which produces a TDS of size at most $(1 + \frac{1}{k})^2 \times |OPT|$ in $O(k^2 n^{2(\lceil 2\sqrt{2}k \rceil)^2})$ time, where OPT is the optimum solution.

Contents

List of Algorithms

List of Figures

List of Abbreviations

APX	The class of polynomial-time constant approximable problems
DS	Dominating set
Dd**DS**	Distance-d dominating set
Dd**IS**	Distance-d independent set
DTIME	Deterministic time
(d, m, ℓ, r) **set**	d-distance m-tuple (ℓ, r)-dominating set
FPTAS	Fully polynomial-time approximation scheme
GMDd**DS**	Geometric minimum distance-d dominating set
GMDd**IS**	Geometric maximum distance-d independent set
GMTDS	Geometric minimum total dominating set
GMVEDS	Geometric minimum vertex-edge dominating set
IS	Independent set
MANET	Mobile ad-hoc network
MDd**DS**	Minimum distance-d dominating set
MDd**IS**	Maximum distance-d independent set
NP	The class of non-deterministic polynomial-time solvable problems
NP-hard	The class of non-deterministic polynomial-time hard problems
OPT	Optimal solution
P	The class of deterministic polynomial-time solvable problems
PTAS	Polynomial-time approximation scheme
TDS	Total dominating set
UDG	Unit disk graph
VC	Vertex cover
VEDS	Vertex-edge dominating set
VLSI	Very large scale integration
WANET	Wireless ad-hoc network

List of Symbols

Symbol	Meaning
$\mathbb{R}^2$	$\{(a,b) \mid a,b \in \mathbb{R}\}$
$\mathcal{P}$	A set of points in $\mathbb{R}^2$
$p \in \mathcal{P}$	p is a member of $\mathcal{P}$
$d(p,q)$	The Euclidean distance between p and q
$w(u,v)$	The weight of the edge uv
$\Delta(P)$	The set of unit disks centered at the points in P
H	A horizontal strip
S_i	A subset of the points lying in H having certain properties
$\mathcal{R}$	A rectangular region in the plane
$\mathcal{M}$	A matrix
ℓ_v	A vertical line
ℓ_h	A horizontal line
$\forall$	For all
$\mid$	Such that
$\mathcal{Q} \subseteq \mathcal{P}$	$\mathcal{Q}$ is a subset of $\mathcal{P}$
$\mathcal{P} \cap \mathcal{Q}$	The intersection of $\mathcal{P}$ and $\mathcal{Q}$
$\mathcal{P} \cup \mathcal{Q}$	The union of $\mathcal{P}$ and $\mathcal{Q}$
$\mathcal{P} \setminus \mathcal{Q}$	The set minus of $\mathcal{P}$ and $\mathcal{Q}$
$\lvert \cdot \rvert$	The cardinality of a set
k	A positive integer
χ	A cell of size $k \times k$
$\sum$	The addition of a sequence of numbers
$\square$	The end of a proof
$\emptyset$	The empty set
$\{.\}$	The set notation
$G = (V,E)$	An undirected connected simple graph with vertex set V and edge set E

$e \in E$	e is a member of E
n	The cardinality of V (or) the cardinality of a point set $\mathcal{P}$
m	The cardinality of E
$d_G(.,.)$	The number of edges on a shortest path between two vertices/sets in G
$N_G(.)$	The open neighborhood of a vertex/set
$N_G[.]$	The closed neighborhood of a vertex/set
$N^d[v]$	The d-distance neighborhood of a vertex v
$N_G[A]$	The closed neighborhood of a set $A \subseteq V$
$K_{1,4}$	The star graph with 5 vertices
ϵ	A positive real number less than 1
ρ	The value of $1 + \epsilon$
η	A single segment in an embedding
$\{\mathcal{A}_\epsilon\}$	A collection of algorithms with ϵ as input parameter
$T(n)$	The time complexity of an algorithm with input size n
$O(\cdot)$	The asymptotic big-oh notation
H_i	A non-empty cell with index i in the hexagonal partition of $\mathcal{R}$
$\lfloor x \rfloor$	The largest integer less than or equal to x
$\lceil x \rceil$	The smallest integer greater than or equal to x
$a \leftarrow b$	Variable a gets the value of b

Chapter 1

Introduction

Dominating set problem is one of the classical combinatorial optimization problem in graph theory. Given a simple graph $G = (V, E)$, a *dominating set* (DS) D is a subset of V such that every vertex $v \in V \setminus D$ is adjacent to at least one vertex u in D. That is, any vertex $v \in V$ is either in D or there exist a vertex $u \in D$ such that $vu \in E$. A graph can have multiple dominating sets (see Figure 1.1). Every graph $G = (V, E)$ has a trivial DS V. The objective of *dominating set problem* is to find a DS of minimum size in a given graph. We call a DS of minimum cardinality as a *minimum dominating set* (MDS). The vertices in D are called as dominators and the rest are called as dominatees. A dominator dominates all its neighbors and itself.

We define a distance-d dominating set (DdDS) problem as follows: for a given integer $d \geq 1$ and a simple undirected graph $G = (V, E)$, a DdDS is a set of vertices $V' \subseteq V$ such that for each vertex $u \in V$, either (i) $u \in V'$, or (ii) there exist a vertex $v \in V'$ which is at most d distance away from u in G. The objective of the *minimum distance-d dominating set* (MDdDS) problem is to find a DdDS of minimum cardinality in a given undirected graph.

Domination and its variants in graphs have many important applications to several areas. The widely used application is routing in wireless ad-hoc networks (WANs) and mobile ad-hoc networks (MANs), for example see [31, 32]. The applications in WANs and MANs are briefly discussed below.

The WANs and MANs do not have any centralized infrastructure such as routers in wired networks, each node (sensor) in the network participate in routing. During message transfer from one node to another, the intermediate nodes act as routers. This causes to reduce the network lifetime as the nodes are battery operated. In MANs, the network topology gets changed frequently based on the mobility of nodes. The classical routing protocols do not work in these networks. The best way to ensure routing is by forming clusters (groups of nodes) and electing a head for each cluster. Cluster heads take the role of routers in data transmission between nodes. If a node moves from one cluster to another, then the node can communicate with the other nodes via its new cluster head. Effective routing can be done by choosing the minimum number of cluster heads. The set of cluster heads is an MDS.

The other well-known applications of DS are in facility location, very large scale integration (VLSI), image processing, etc. In facility location problems, a set of facilities (service providers) such as hospitals, fire stations, etc., and a set of clients are given. For any client, the maximum distance to reach a facility is predefined. The objective is to open a subset of facilities such that every client is served by at least one facility. Opening a facility involves a certain cost. Hence, we are interested in opening as minimum facilities as possible subject to every client is served by at least one facility. In this case, opening facilities in an MDS solves the purpose. Some examples of facility location problems in geometric intersection graphs can be found in [8, 9, 49] and competitive facility location problem in the form of Voronoi games can be found in [85].

An *independent set* (IS) of a graph $G = (V, E)$ is a set of vertices $V' \subseteq V$ such that no two vertices in V' are adjacent in G (see the red vertices in Figure 1.1 (b), and (c)). The objective of the *independent set problem* for a given graph G is to find an independent set of maximum cardinality. We call an IS of maximum cardinality as a maximum independent set (MIS) or *largest independent set*. A *maximal independent set* of G is an independent set which is not a proper subset of any other independent set of G. An *independent dominating set* is a dominating set which is also an independent set.

For an integer $d \geq 2$, a *distance-d independent set* (DdIS) of an undirected graph

$G = (V, E)$ is an independent set I of G such that the shortest path distance (i.e., the number of edges on a shortest path) between every pair of vertices in I is at least d. For a given undirected graph G, the objective of the maximum distance-d independent set problem is to find a DdIS of maximum cardinality in G. A DdIS of maximum cardinality is called as *maximum distance-d independent set* (MDdIS). Observe that the DdIS problem is a generalized version of the MIS problem. In fact, for $d = 2$, the DdIS problem and MIS problem are the same. The well-known applications of the maximum independent set are map labelling [101], information/coding theory [14], computer vision, and scheduling etc.

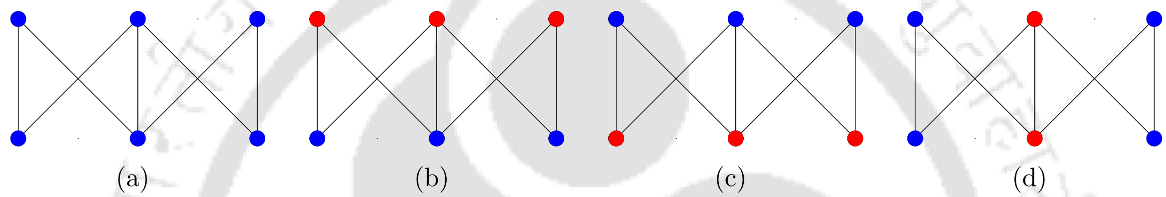


Figure 1.1: (a) An example graph, (b) a DS, (c) an other DS, and (d) an MDS

We define a *d-distance m-tuple (ℓ, r)-dominating set* problem in graphs as follows: given a simple undirected graph $G = (V, E)$, and positive integers d, m, ℓ and r, a subset $V' \subseteq V$ is said to be a d-distance m-tuple (ℓ, r)-dominating set ((d, m, ℓ, r) set) if it satisfies the following two conditions: (i) each vertex $v \in V$ is d-distance dominated by at least m vertices in V', and (ii) each r size subset $U \subseteq V$ is d-distance dominated by at least ℓ vertices in V'. Here, a vertex v is d-distance dominated by another vertex u means the shortest path distance between u and v is at most d in G. A set U is d-distance dominated by a set of ℓ vertices means size of the union of the d-distance neighborhood of all vertices of U in V' is at least ℓ. The objective of the d-distance m-tuple (ℓ, r)-domination problem is to find a minimum size subset $V' \subseteq V$ such that V' is a (d, m, ℓ, r) set of the graph G. If $m \geq \ell$, then the second condition in the definition of (d, m, ℓ, r) set is redundant. In the case of $m = \ell (= k$, say), the (d, m, ℓ, r) set is known as k-tuple dominating set in the literature. Note that, if $m = \ell$ then the value of $r > 1$ is irrelevant. Therefore, we assume $r = 1$ in case of $m = \ell$. From now onwards, we assume that $m < \ell$. If $d = 1, m = 2, \ell = 3, r = 2$ then (d, m, ℓ, r) set is known as a liar's dominating set in the literature. The objective of the d-distance m-tuple (ℓ, r)-domination problem is to find a minimum size d-distance m-tuple (ℓ, r)-dominating set in a given graph G, and we call this problem as the *minimum (d, m, ℓ, r) set* problem. In Figure 1.2, the set of vertices $\{e, f, i\}$ form a 3-distance 2-tuple $(3, 4)$-dominating set for the graph.

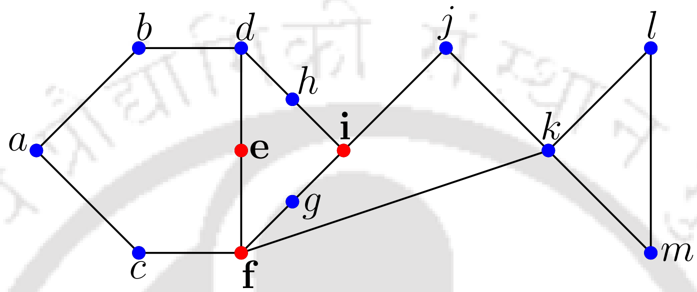


Figure 1.2: $\{e, f, i\}$ is a 3-distance 2-tuple $(3, 4)$-dominating set.

The problem has important applications such as fault tolerance in wireless/sensor networks. One specific real-time application for $r = 2$ is as follows: suppose that in a graph $G = (V, E)$ each vertex is a possible location for an intruder such as a thief, a saboteur, a fire, or some possible fault. Assume also that there is exactly $\min(\ell - m, \lceil \ell/2 \rceil - 1)$ intruders in the system represented by G. A protection device placed at a vertex v is assumed to be able to (i) detect the intruder at any vertex in its d-distance neighborhood $N_G^d[v]$, and (ii) report the vertex $u \in N_G^d[v]$ at which the intruder is located. We are interested in deploying protection devices at a minimum number of vertices so that the intruder can be detected and identified correctly. This can be solved by finding a minimum cardinality m-tuple dominating set, say D, of G and deploying protection devices at all the vertices of D. If any one protection device can fail to detect the intruder, then to correctly detect and identify the intruder one needs to place the protection devices at all the vertices of a minimum cardinality $2m$-tuple dominating set of G. Now it may so happen that all the protection devices detect the intruder location correctly but while reporting some of these protection devices can misreport or lie (either deliberately or through a transmission error) about the intruder location. Assume that at most $\min(\ell - m, \lceil \ell/2 \rceil - 1)$ protection devices in the d-distance neighborhood of an intruder location can lie. Under these circumstances, to protect the network we have to install the protection devices at all the vertices of a minimum d-distance m-tuple (ℓ, r) dominating set.

A *vertex-edge dominating set* (VEDS) of a simple undirected graph $G = (V, E)$ is a set $D \subseteq V$ of G such that every edge of G is incident with a vertex of D or a vertex adjacent to a vertex of D. The *VEDS problem* asks to find a VEDS of minimum size in a given graph. In Figure 1.3, the set of vertices $\{c, d\}$ form a vertex-edge dominating set for the graph.

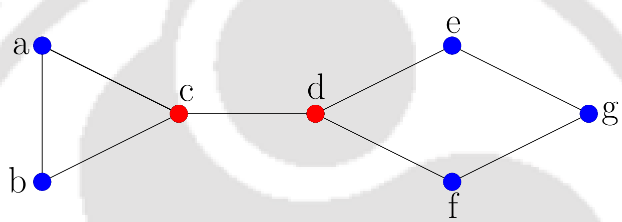


Figure 1.3: $\{c, d\}$ is a vertex-edge dominating set.

One of the real-time application of the VEDS problem is as follows: consider a client-server architecture based model in a computer networking system. A client can communicate with a server either directly or through some intermediate clients and servers and vice versa. To secure the communication of the whole networking system, one has to secure each and every communication line between any two adjacent pairs of clients and servers. We refer a communication line between any two adjacent pair of clients or servers is secure if and only if one of them is a secure agent or any one of its open neighborhood is a secure agent. A secure agent can secure the communication line not only with the clients or servers who are directly linked to them but also to the clients or servers who are at a distance two from them. As the deployment of secure agents are too costly we need to minimize the deployment of secure agents in the networking system. The smallest group of secure agents with this property is a minimum vertex-edge dominating set for the graph which represents the computer network.

One of the variants of dominating set problem is *total dominating set* and is defined as follows: let $G = (V, E)$ be an undirected graph. We call $D_t \subseteq V$ as a total dominating set (TDS) of G if each vertex $v \in V$ has a dominator in D_t other than itself. Therefore, a vertex $v \in D$ (dominating set) dominates all its neighbors and itself whereas a vertex $v \in D_t$ (total dominating set) dominates all its neighbors other than itself. The objective of the TDS problem is to find a minimum size subset $D_t \subseteq V$ such that D_t dominates all the vertices in V. In Figure 1.4, the set of vertices $\{c, d\}$ form a total dominating set for the graph.

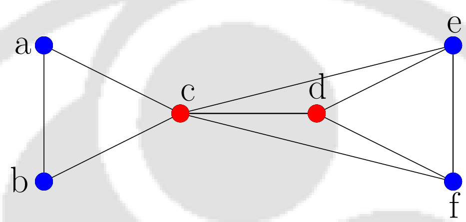
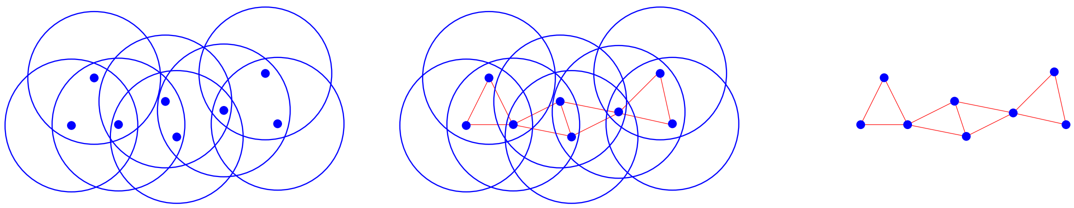


Figure 1.4: $\{c, d\}$ is a total dominating set.

One of the applications of the problem is identifying the location of monitoring devices, such as surveillance cameras or fire alarms, to safeguard a system that can be modeled by total domination in graphs. The problem of placing monitoring devices in a system, where each monitoring device can be placed in such a way that every site in the system (including the monitors themselves) is adjacent to one of the monitoring devices. In this case, placing the monitoring devices in each solution point of the total domination of the system solves the problem.

Many other variants of dominating set can be found in the book *Fundamental of Domination in Graphs* by Haynes et al. [50].

An algorithm for an optimization (minimization or maximization) problem is said to be a *ρ-factor approximation algorithm* if for every instance of the problem the algorithm produces a feasible solution whose value is within a factor ρ of the optimal solution value and runs in polynomial-time of the input size. Here, ρ is called the approximation ratio or approximation factor of the algorithm.



(a) (b) (c)

Figure 1.5: (a) A family of unit disks in the plane, (b) the corresponding unit disk graph, and (c) the final graph without disks

A *polynomial-time approximation scheme* (PTAS) for an optimization problem is a collection of algorithms $\{\mathcal{A}_\epsilon\}$ such that for a given $\epsilon > 0$, $\mathcal{A}_\epsilon$ is a $(1+\epsilon)$-factor approximation algorithm in case of minimization problem ($(1-\epsilon)$ in case of maximization). The running time of $\mathcal{A}_\epsilon$ is required to be polynomial in the size of the input instance of the problem depending on ϵ. If the running time of $\mathcal{A}_\epsilon$ is polynomial in the size of the problem and in $\frac{1}{\epsilon}$, then $\{\mathcal{A}_\epsilon\}$ is said to be *fully polynomial-time approximation scheme* (FPTAS). Note that for a problem existence of FPTAS implies PTAS, but the converse need not be true.

The class APX (an abbreviation of approximable) is the set of optimization problems in NP that admit constant factor approximation algorithms. A problem is said to be APX-hard if every problem in APX is reducible to that problem via PTAS reduction. PTAS reduction from an optimization problem A to a problem B means, a PTAS for B can be composed with the reduction to obtain a PTAS for the problem A. A problem is said to be APX-complete if it belongs to both APX and APX-hard.

Williamson and Shmoys [104] and Vazirani [99] are excellent references to start in the field of approximation algorithms for researchers who are interested in designing approximation algorithms for optimization problems.

An *intersection graph* of objects is a graph, where the vertex set is the set of objects and there is an edge between two objects if their intersection is non-empty. A *unit disk graph* (UDG) is an intersection graph of a family of disks of equal radii in the plane.

Given a family $D = \{d_1, d_2, \ldots, d_n\}$ of n equiradius disks in the plane, the corresponding UDG $G = (V, E)$ is defined as follows: each vertex $v_i \in V$ corresponds to the center of the disk $d_i \in D$, and there is an edge between two vertices if and only if the Euclidean distance between the corresponding disk centers is at most 1 i.e., the corresponding disk centers lie one in the other disk (see Figure 1.5).

The network topology of WANs can be modelled as a unit disk graph $G = (V, E)$ [3, 10], there is an edge between two nodes in the network if and only if one is in the range of the other.

1.1 Scope of the Thesis

The optimization problems such as maximum distance-d independent set (MDdIS) problem, minimum distance-d dominating set (MDdDS) problem, minimum d-distance m-tuple (ℓ, r)-dominating set (minimum (d, m, ℓ, r) set) problem, minimum vertex-edge dominating set (MVEDS) problem, and minimum total dominating set (MTDS) problem have many real-world applications. All these problems belong to NP-hard class and none of them admit constant factor approximation algorithms for general graphs, unless P = NP. This motivated many researchers to study the problems for other graph classes. We study the d-distance m-tuple (ℓ, r)-dominating set problem for general graphs and all the other four problems for unit disk graphs with known disk position and focus on designing constant factor approximation algorithms and polynomial-time approximation schemes that outperform the existing results in the literature.

1.2 Organization of the Thesis

The thesis contains seven chapters and is organized as follows.

Chapter 2 : Literature Review. In this chapter, we discuss the existing literature related to the problems considered in this thesis.

Chapter 3 : Generalized Maximum Independent Set and Minimum Dominating Set Problem.

In this chapter, we prove that the maximum distance-d independent set problem and minimum distance-d dominating set problem belong to the NP-hard class in unit disk graphs. We propose constant factor approximation algorithms and PTASes for both problems.

Chapter 4 : Minimum d-Distance m-Tuple (ℓ, r)-Dominating Set Problem.

In this chapter, we prove that the 1-distance m-tuple (ℓ, r)-dominating set problem and the d-distance m-tuple $(\ell, 2)$-dominating set problem belong to NP-hard class in graphs. We also prove that for any $\varepsilon > 0$, the 1-distance m-tuple (ℓ, r)-domination problem and the d-distance m-tuple $(\ell, 2)$-domination problem cannot be approximated within a factor of $(\frac{1}{2} - \varepsilon) \ln |V|$ and $(\frac{1}{4} - \varepsilon) \ln |V|$, respectively, unless P = NP.

Chapter 5 : Minimum Vertex-Edge Dominating Set Problem.

In this chapter, we study hardness result of the problem. We propose a constant factor approximation algorithm and a PTAS for the minimum vertex-edge dominating set problem.

Chapter 6 : Minimum Total Dominating Set Problem.

In this chapter, we prove the problem belongs to the NP-hard class in unit disk graphs, propose an almost linear-time 8-factor approximation algorithm, and a PTAS for the problem.

Chapter 7 : Conclusion and Future Work. In this chapter, we summarize the work done in this thesis and make concluding remarks. We also identify a number of open problems as future research work.

Chapter 2

Literature Review

In this chapter, we discuss the state of the art results for all the considered problems, namely, generalized maximum independent set and minimum dominating set, d-distance m-tuple (ℓ, r)-dominating set, minimum vertex-edge dominating set, and minimum total dominating set, in general graphs, unit disk graphs, and other intersection graphs.

For an integer $d \geq 2$, a *distance-d independent set* (DdIS) of an undirected graph $G = (V, E)$ is an independent set I of G such that the shortest path distance (i.e., the number of edges on a shortest path) between every pair of vertices in I is at least d. For a given undirected graph G, the objective of the maximum distance-d independent set problem is to find a DdIS of maximum cardinality in G. A DdIS of maximum possible size is called as *maximum distance-d independent set* (MDdIS).

The maximum independent set (MIS) problem is known to be NP-hard for general graphs [43] including many sub-class of planar graphs, namely, planar graphs of maximum degree 3 [42], planar graphs of large girth [75], cubic planar graphs [44], triangle-free graphs [83], $K_{1,4}$-free graphs [72], etc.

Tarjan and Trojanowski [93] presented a naive algorithm for finding MIS in a graph having n-vertices in $O(2^{\frac{n}{3}})$ time. Later Robson [88] improved the complexity to $O(2^{0.276n})$. Xiao and Nagamochi [105] gave a better bound for finding the MIS problem in $1.1996^n n^{O(1)}$ time and in polynomial space. For the graphs with maximum degree 6 and 7, they gave algorithms to find MIS, which run in $1.1893^n n^{O(1)}$ time and $1.1970^n n^{O(1)}$ time, respectively. Johnson et al. [61] presented an algorithm, which produces lexicographic ordering of all maximal independent sets of a graph having a polynomial-delay between two successive independent sets with exponential space complexity. In that paper, they also proved that there is no such polynomial-delay algorithm exists for generating all maximal independent sets in reverse lexicographic order unless P=NP.

Andrade [5] gave the first local search algorithm for finding an independent set of a graph. Later the problem studied on pseudo-disks in the plane by Chan and Har-Peled [17]. They analyzed the problem for both weighted and unweighted cases and gave a PTAS via local-search algorithm for the unweighted case and for the weighted case, they gave a constant-factor approximation by an LP-based rounding scheme.

In general, the MIS problem cannot be approximated within a constant factor unless P=NP [6]. However, the problem is polynomially solvable for bipartite graphs, outerplanar graphs, perfect graphs, claw-free graphs, chordal graphs [45, 58]. The MIS problem is well studied in UDGs too and is shown to be NP-hard [25]. Unlike in general graphs, the problem admits approximation algorithms and approximation schemes in UDGs. Marathe et al. [70] proposed a 3-factor approximation algorithm for MIS problem in UDGs, which is later improved by Halldórsson [47] to $2.5 + \epsilon$. Matsui [71] proposed a $(1 - 1/r)$-factor approximation algorithm for the MIS problem in UDGs in $O(rn^4 \lceil 2(r-1)/\sqrt{3} \rceil)$ time. Erlebach et al. [37] proposed a PTAS for the problem in UDGs. Nieberg et al. [77] presented a PTAS for the maximum weight independent set problem in unit disk graphs, where the geometric representation of the unit disks does not require. They also proposed a robust approximation algorithm that accepts any graph as input and either returns a $(1 + \epsilon)$-approximate independent set or a certificate showing that the input graph is not a unit disk graph. Das et al. [30] proposed a 2-factor approximation algorithm for the MIS problem in unit disk graph in $O(n^3)$ time. They also proposed a PTAS for the problem in UDGs, which produces a solution of size $\frac{1}{(1+\frac{1}{k})^2}|OPT|$ in $O(k^4 n^{\sigma_k \log k} + n \log n)$ time, where OPT is the optimal solution of the problem, $k > 1$ is an integer, and $\sigma_k \leq \frac{7k}{3} + 2$. Recently, Das et al. [29] proposed a 2.16-factor approximation algorithm in $O(n \log^2 n)$ time and a 2-factor approximation algorithm in $O(n^2 \log n)$ time for the MIS problem in UDGs. They also proposed a PTAS for the same problem.

The distance-d independent set (DdIS) problem, for any fixed $d \geq 3$, is known to be NP-hard for bipartite graphs [18] and planar bipartite graphs of maximum degree 3 [38]. It is also known that getting an $n^{\frac{1}{2}-\epsilon}$-factor approximation result, for any $\epsilon > 0$, on bipartite graphs is NP-hard (this result also holds for chordal graphs when $d \geq 3$ is an odd number) [38]. The problem is polynomially solvable for some intersection graphs, such as interval graphs, trapezoid graphs, and circular-arc graphs [1]. If the input graph is restricted to be a chordal graph, then the problem is solvable in polynomial time for any even $d \geq 2$; on the other hand, the problem belongs to the NP-hard class for any odd $d \geq 3$ [38]. Eto et al. [39] studied the problem on r-regular graphs and planar graphs. The authors showed that for $d \geq 3$ and $r \geq 3$, the DdIS problem on r-regular graphs is APX-hard, and proposed $O(r^{d-1})$ and $O(\frac{r^{d-2}}{d})$-factor approximation algorithms. When $d = r = 3$, they enhanced their $O(\frac{r^{d-2}}{d})$-factor result to a 2-factor approximation result (later, the approximation factor is improved to 1.875 [40]). Finally, they proposed a PTAS in the case of planar graphs. Montealegre and Todinca studied the problem in graphs with few minimal separators [74].

A DdDS for an integer $d \geq 1$ in a simple unweighted graph $G = (V, E)$ is defined as a set of vertices $V' \subseteq V$ such that, for each vertex $u \in V$, either (i) $u \in V'$, or (ii) $v \in V'$ such that, the shortest path distance between u and v is at most d. The objective of the *minimum distance-d dominating set* (MDdDS) problem is to find a DdDS of minimum cardinality in a given graph G.

The minimum dominating set (MDS) problem is known to be NP-hard [43]. Raz and Safra [87] proved the inapproximability for the MDS problem by showing that there does not exist any approximation algorithm better than $O(\log n)$-factor approximation algorithm unless P=NP. Due to the lack of scope in better approximation results in general graphs, researchers tried the geometric version of the MDS problem to get a better approximation factor.

The MDS problem is studied in UDGs and proved to be NP-hard [25]. Nieberg and Hurink [76] proved that the problem admits a $(1+\epsilon)$-factor approximation algorithm (PTAS) for $0 < \epsilon \leq 1$ in $n^{O(1/\epsilon \log 1/\epsilon)}$ time. By assigning $\epsilon = 1$, a 2-approximation algorithm can be obtained, which is fastest. The running time of this algorithm is $O(n^{81})$ [33]. Gibson and Pirwani [46] gave a PTAS for MDS problem of arbitrary size disk graph, which runs in $n^{O(\frac{1}{\epsilon^2})}$ time.

For the MDS problem in unit disk graphs, a 5-factor approximation algorithm is proposed by Marathe et al. [70] in $O(n^2)$ time. Carmi et al. [16] proposed a 5-factor approximation algorithm for the MDS problem in arbitrary radius size disk graph. Fonseca et al. [41] improved the factor to $\frac{44}{9}$ for MDS problem in unit disk graph by using the local improvement technique, which runs in $O(n \log n)$ time. De et al. [33] proposed a 12-factor approximation algorithm for the MDS problem in unit disk graph with running time $O(n \log n)$. In the same paper, they proposed a 4-factor and 3-factor approximation algorithm for the MDS problem in time $O(n^8 \log n)$ and $O(n^{15} \log n)$ respectively. Carmi et al. [15] improved the time complexity of the 4-factor approximation algorithm to $O(n^6 \log n)$. They also proposed a simple 5-factor approximation algorithm in $O(n \log k)$ time, where k is the size of the output. In the same paper, they also proposed $\frac{14}{3}$-factor, 3-factor and $\frac{45}{13}$-factor approximation algorithm for MDS problem in unit disk graphs with time complexity $O(n^5 \log n)$, $O(n^{11} \log n)$ and $O(n^{10} \log n)$ respectively. Finally, with the help of shifting lemma, they proposed a $\frac{5}{2}$-factor approximation algorithm in $O(n^{20} \log n)$ time.

Given a simple undirected graph $G = (V, E)$, and positive integers d, m, ℓ and r, a subset $V' \subseteq V$ is said to be a d-distance m-tuple (ℓ, r)-dominating set if it satisfies the following conditions: (i) each vertex $v \in V$ is d-distance dominated by at least m vertices in V', and (ii) each r size subset U of V is d-distance dominated by at least ℓ vertices in V'. Here, a vertex v is d-distance dominated by another vertex u means the shortest path distance between u and v is at most d in G. A set U is d-distance dominated by a set of ℓ vertices means size of the union of the d-distance neighborhood of all vertices of

U in V' is at least ℓ. The objective of the d-distance m-tuple (ℓ, r)-domination problem is to find a minimum size subset $V' \subseteq V$ satisfying the above two conditions.

The domination problem is one of the most studied problems in the literature for its wide range of applications. The concepts of dominations and its variations are widely studied and can be seen in [50, 51].

One of the variations of domination is the k-tuple domination problem and was introduced by Harary and Haynes [48]. When $k = 1$, it is the usual domination problem. For $k = 2$, it is called double domination [48]. In the same paper, they discuss exact values of the double domination numbers for some special graphs and various bounds of the double and the k-tuple domination numbers in terms of other parameters. The hardness results and bounds for the k-tuple domination number for various sub-classes of graphs can be found in [69, 86].

In 2009, Slater [91] first introduced the 1-distance 2-tuple (3,2) domination problem known as the liar's dominating set (LDS) problem in the literature. The author of the article proved that the problem is NP-hard for general graphs and proposed various bounds for graphs, trees, cycles, and paths. The problem is also studied for different sub-classes of graphs and proved to be NP-hard for bipartite graphs [89], split graphs and chordal graphs [78], doubly chordal graphs [80], whereas polynomial-time solvable in trees [78], block graphs [80], proper interval graphs [79]. Panda et al. [80] studied the approximability of the problem and gave an $O(\ln \Delta)$-factor approximation algorithm, where Δ is the degree of the given graph. Alimadadi et al. [2] provided the characterization of graphs and trees for which the LDS cardinality is $|V|$ and $|V| - 1$, respectively.

A set $D \in V$ is called a vertex-edge dominating set of $G = (V, E)$ if for each edge $e = uv \in E$, either u or v is in D or one vertex from their neighbor is in D. Simply, a vertex $v \in V$, vertex-edge dominates every edge uv, as well as every edge adjacent to these edges. The vertex-edge dominating problem is to find a minimum vertex-edge dominating set of G.

The vertex-edge dominating set problem was introduced by Peters [81] and then studied further by different researchers. In particular, bounds on the vertex-edge domination number in several graph classes were studied in [12, 65, 67, 68, 94], vertex-edge degrees and vertex-edge domination polynomials of different graphs were discussed in [20, 57, 102, 103], whereas the relations between some vertex-edge domination parameters were discussed in [12, 21, 63, 67, 68], several algorithmic aspects were discussed in [67]. Some variants of vertex-edge domination problem were studied in [11, 22, 62, 64, 90].

The minimum cardinality of a vertex-edge dominating set (double vertex-edge dominating set, respectively) of G is termed the *vertex-edge domination number* and denoted by $\gamma_{ve}(G)$ (the *double vertex-edge domination number*, $\gamma_{dve}(G)$, respectively). Krishnakumari et al. [65] proved that for every tree T of order $n \geq 3$ with ℓ leaves and s support vertices, we have $\frac{(n-\ell-s+3)}{4} \leq \gamma_{ve}(T) \leq \frac{n}{3}$. In [62], Krishnakumari et al. showed that determining $\gamma_{dve}(G)$ for bipartite graphs is NP-hard, whereas for every non-trivial connected graphs G, $\gamma_{dve}(G) \geq \gamma_{ve}(G) + 1$, and for every tree T, we have $\gamma_{dve}(T) = \gamma_{ve}(T) + 2$. They also provided two lower bounds on the double vertex-edge domination number of trees and unicycle graphs in terms of order n, the number of leaves, and support vertices, respectively.

The minimum VEDS problem cannot be approximated within a constant factor unless NP$\subset$ DTIME($|V|^{O(\ln \ln |V|)}$) [67]. Boutrig et al. [12] presented a new relationship between the vertex-edge domination and some other domination parameters, answering the four open questions posed by Lewis [67]. Then, for every non-trivial connected $K_{1,k}$-free graph, with $k \geq 3$, they provided an upper bound for the independent vertex-edge domination number in terms of the vertex-edge domination number and showed that for every non-trivial tree the independent vertex-edge domination number can be bounded by the domination number. For connected C_5-free graphs, they also established an upper bound on the vertex-edge domination number. Next Boutrig and Chellali [11] studied the total vertex-edge domination. The minimum cardinality of a total vertex-edge dominating set of graph G called the *total vertex-edge domination number* and denoted by $\gamma^t_{ve}(G)$. They showed that determining $\gamma^t_{ve}(G)$ for bipartite graphs is NP-hard, and in case of tree T different from a star having order n, with ℓ leaves and

s support vertices, we have $\gamma_{ve}^t(G^T) \leq \frac{(n-\ell+s)}{2}$. In the same article, they established a necessary condition for a graph G to satisfy $\gamma_{ve}^t(G) = 2\gamma_{ve}(G)$ and for a tree T, $\gamma_{ve}^t(T) = 2\gamma_{ve}(T)$.

Later Venkatakrishnan and Kumar [100] proved that the minimum double vertex-edge dominating set problem is NP-hard for chordal graphs and APX-hard for bipartite graphs with maximum degree 5. They also proposed a linear-time algorithm for finding a minimum double vertex-edge dominating set in proper interval graphs. In addition, they showed that the minimum double vertex-edge dominating set problem can not be approximated the factor $(1-\epsilon)\ln|V|$ for any $\epsilon \geq 0$ unless NP$\subset$ DTIME($|V|^{O(\ln\ln|V|)}$). Finally, the influence of edge removal, edge addition, and edge subdivision on the double vertex-edge domination number of a graph was investigated by Krishnakumari and Venkatakrishnan [63]. Horoldagva et al. [57] obtained some results on the regularity and irregularity of vertex-edge and edge-vertex degrees in graphs. Recently, Żyliński [106] proved that for any connected graph G of order $n \geq 6$, $\gamma_{ve}(G) \leq \left\lfloor \frac{n}{3} \right\rfloor$.

Let $G = (V, E)$ be an undirected graph. We call $D_t \subseteq V$ as a total dominating set (TDS) of G if for each vertex $v \in V$ has a dominator (i.e., neighbor) in D other than itself. Here we consider the TDS problem in unit disk graphs, where the objective is to find a minimum cardinality total dominating set for an input graph.

In 1980, Cockayne et al. [26] introduced the total domination problem and proved that for any connected graph G of $n(\geq 3)$ vertices the cardinality of minimum total dominating set, denoted by λ_t, is less than $\frac{2}{3}n$ i.e., $\lambda_t \leq \frac{2}{3}n$. Brigham et al. [13] proved that the total domination number is exactly $\frac{2}{3}n$ for the connected graph G of order $n(\geq 3)$, where G is either C_3 (cycle graph of 3 vertices), C_6 or 2-corona of some connected graph. Sun [92] proved the bound to $\lambda_t \leq \left\lfloor \frac{4}{7}(n+1) \right\rfloor$, for connected graphs having order n with minimum degree at least 2. Chvátal and McDiarmid [24] and Tuza [96] independently proved a theorem concerning transversals in hypergraphs, which gives a bound on total domination number. The bound is $\lambda_t \leq \frac{n}{2}$ for the graphs with order n and minimum degree at least 3. For the graphs with minimum degree at least 4, Thomassé and Yeo [95] proposed a result for hypergraphs, which bounds the total domination number by $\lambda_t \leq \frac{3}{7}n$. In [34], DeLaViņa et al. proved that the total domination number of any connected graph is equal to the total domination number of a spanning tree of the same graph. Another interesting aspect of trees with respect to total domination is that it is possible to characterize some vertices that are in every total dominating set or not in any total dominating set [27]. Furthermore, Haynes and Henning established three equivalent conditions for a tree to have a unique minimum total dominating set [52]. Chellali and Haynes [19] proved $\lambda_t \geq \frac{n+2-\ell}{2}$ for a nontrivial tree of n vertices with ℓ leaves. Dorfling et al. [36] bound the total domination number of planar graphs having different diameter and radius. Pfaff et al. [82] showed that computing λ_t for general graphs is NP-complete. In the same paper, they also showed that calculating λ_t for bipartite graphs remains NP-complete. However, a linear-time algorithm exists for computing λ_t in tree graph [66]. The total domination number in the case of star graphs, complete graphs, binary star graphs, and complete bipartite graphs is 2 [4]. In the same article, they have observed that for cycles and paths, the total domination number can be calculated in polynomial time. They have also established a set of relation between (i) λ_t and the maximum degree, and (ii) λ_t and the cut vertices of the graph. See [4, 50, 51, 53, 54] for detail survey on the TDS problem.

Chapter 3

Generalized Independent and Dominating Set Problems in Unit Disk Graphs

The *independent set* and *dominating set* problems are well known classical combinatorial optimization problems in graph theory due to their many important applications, including but not limited to networks, map labeling, scheduling, clustering, facility location, etc. [97].

In this chapter, we study the geometric version of the maximum distance-d independent set and minimum distance-d dominating set problem.

Definition 3.0.1. (Geometric maximum distance-d independent set (GMDdIS) problem) *Given an unweighted unit disk graph $G = (V, E)$ corresponding to a point set $P = \{p_1, p_2, \ldots, p_n\}$ for disk centers in the plane and an integer d, find a maximum cardinality subset $I \subseteq V$ such that for every pair of vertices $p_i, p_j \in I$ $(1 \leq i, j \leq n)$ the length (number of edges) of the shortest path between p_i and p_j in G is at least d.*

Definition 3.0.2. (Geometric minimum distance-d dominating set (GMDdDS) problem) *Given a simple unit disk graph $G = (V, E)$ corresponding to a point set $P = \{p_1, p_2, \ldots, p_n\}$ for disk centers in the plane and an integer d, find a minimum* *cardinality subset $D \subseteq V$ such that for each vertex $p_i \in V$, either $p_i \in D$ or there exist a vertex $p_j \in D$ such that, the length (number of edges) of the shortest path between p_i and p_j in G is at most d.*

- The goal of this chapter is to prove hardness result for GMDdIS problem as well as hardness result for GMDdDS problem. We also propose a 4-factor approximation algorithm and a PTAS for both the GMDdIS problem and GMDdDS problem.

3.1 The Hardness Result of GMDdIS Problem

In this section, we prove that the decision version of the GMDdIS problem belongs to the class NP-complete. For a fixed integer $d \geq 3$, the decision version D-GMDdIS of the GMDdIS problem is defined as follows:

Given a unit disk graph $G = (V, E)$ defined on a point set P, an integer d, and a positive integer $k \leq |V|$, does there exist a distance-d independent set of size at least k in G?

Lemma 3.1.1. *The problem belongs to the class NP.*

Proof. Given any subset $V' \subseteq V$, we can verify whether each pair of vertices in V' is distance-d independent or not in polynomial-time using Floyd-Warshall's all-pair shortest path algorithm [28]. □

Definition 3.1.2. (Girth) The length of the smallest cycle in a graph is known as girth.

Now, to show D-GMDdIS ($d \geq 3$) problem belongs to the NP-hard class, we need to prove that the distance-d independent set problem in planar bipartite graphs with maximum degree 3 and girth at least d is NP-hard.

Lemma 3.1.3. *The distance-d independent set problem in planar bipartite graphs with maximum degree 3 and girth at least d is NP-hard.*

Proof. In [38], it was shown that the distance-d independent set problem in planar bipartite graphs with maximum degree 3 belongs to NP-hard class using polynomial-time reduction of it from the distance-2 independent set problem in planar cubic graphs, which is known to be in NP-hard class [60]. The reduced graph in their reduction has girth at least d and hence the distance-d independent set problem in planar bipartite graphs with maximum degree 3 and girth at least d is NP-hard. □

Now, we show that the D-GMDdIS ($d \geq 3$) problem belongs to the NP-hard class by polynomial-time reduction of D-GMDdIS from *distance-d independent set problem ($d \geq 3$) in planar bipartite graphs with girth at least d and maximum degree 3*, which is known to be NP-hard [38]. The decision version of DdIS problem in planar bipartite graphs is defined as follows:

> *Given an unweighted planar bipartite graph $G = (V, E)$ with girth at least d and maximum vertex degree 3, and a positive integer $k \leq |V|$, does there exist a distance-d independent set of size at least k in G?*

Our reduction is based on the concept of planar embedding of planar graphs. The concept of grid drawings of the plane graphs also similar to the concept of planar embeding of planar graphs and can be found in [73, 84]. The following lemma is very useful in our reduction.

Lemma 3.1.4 ([98]). *A planar graph $G = (V, E)$ with maximum degree 4 can be embedded in the plane using $O(|V|^2)$ area in such a way that its vertices are at integer coordinates and its edges are drawn using axis-parallel line segments at integer coordinates (i.e., edges lie on the lines $x = i_1, i_2, \ldots$ and/or $y = j_1, j_2, \ldots$, where $i_1, i_2, \ldots, j_1, j_2, \ldots$ are integers).*

Corollary 3.1.5. *Let $G = (V, E)$ be a planar bipartite graph with maximum degree 3 and girth at least d ($d \geq 3$). G can be embedded on a grid in the plane, whose each grid cell is of size $d \times d$, so that its vertices lie at points of the form $(i * d, j * d)$ and its edges are drawn using a sequence of consecutive line segments drawn on the vertical lines of* *the form $x = i * d$ and/or horizontal lines of the form $y = j * d$, for some integers i and j* (see Figure 3.1).

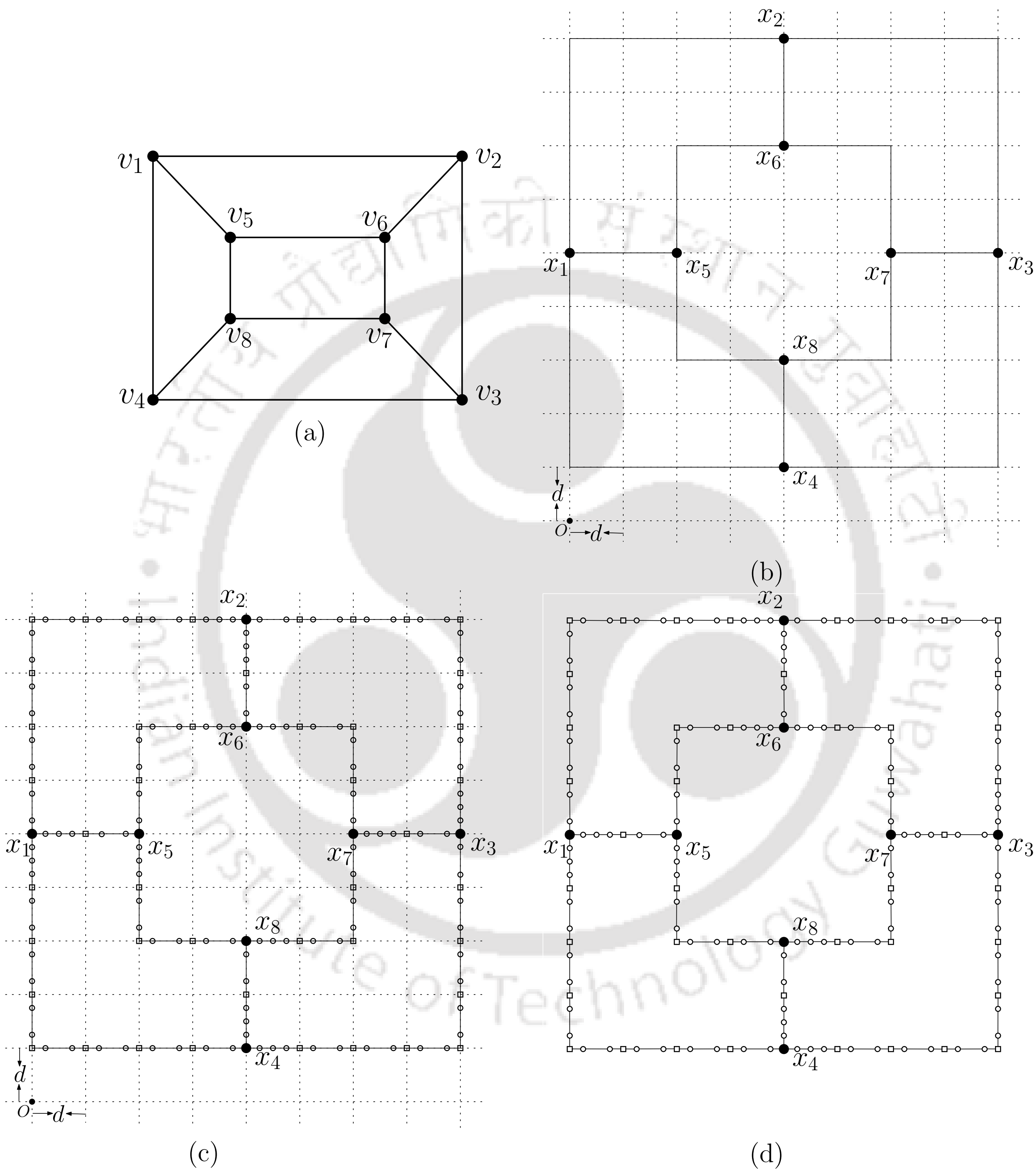


Figure 3.1: (a) A planar bipartite graph G of maximum degree 3, (b) its embedding G' on a grid of cell size 3×3, (c) adding of extra points to G', and (d) obtained UDG G''.

Proof. Lemma 3.1.4 suggests that, any planar graph G of maximum degree 4 can be embedded on a grid in the plane so that;

i Each vertex v_i of G is associated with a point with integer coordinates in the plane.

ii An edge of G is represented as a sequence of alternating horizontal and/or vertical line segments drawn on the grid lines. For example, see edges v_1v_4 or v_2v_6 in Figure 3.1(a). The edge v_1v_4 is drawn as a sequence of four vertical line segments and four horizontal line segments in the embedding (see x_1x_4 in Figure 3.1(b)). Similarly, the edge v_2v_6 is drawn as a sequence of two vertical line segments in the embedding.

iii No two sets of consecutive line segments correspond to two distinct edges of G have a common point unless the edges incident at a vertex in G.

□

This kind of embedding is known as *orthogonal drawing* of a graph.

Lemma 3.1.6. *[7] For a graph $G = (V, E)$, an orthogonal drawing of the graph with at most 2 bends along each edge can be produced in linear time.*

Let $G = (V, E)$ be an arbitrary instance of DdIS for planar bipartite graph having maximum degree three and girth at least d. Let $V = \{v_1, v_2, \ldots, v_n\}$ and $E = \{e_1, e_2, \ldots, e_m\}$. We denote the shortest path distance between two vertices v_i and v_j in G by $d_G(v_i, v_j)$ and v_i, v_j are said to be distance-d independent in G if and only if $d_G(v_i, v_j) \geq d$.

We construct a graph $G' = (V', E')$ by embedding G on a grid in which each cell is of size $d \times d$ as described in Corollary 3.1.5. Let $V' = \{x_1, x_2, \ldots, x_n\}$ be the vertices in G' corresponding to $v_1, v_2, \ldots, v_n$ in G. The coordinate of each member in V' is of the form $(d * i, d * j)$, where i, j are integers, and shown using big dots in Figure 3.1(c). Let ℓ be the number of line segments used for drawing all the edges in G'. To make G' a UDG we introduce a set Y of extra points on the segments used to draw the edges of G'.

Thus, the set of points in V' (hereafter denoted by X) together with Y form a UDG G''. Let $x_i x_j$ be an edge in G' corresponding to the edge $v_i v_j$ in G and has ℓ' grid segments. We introduce $\ell' * d$ points on the polyline denoting the edge $x_i x_j$ in such a way that (i) after adding the extra points, the length of the path from x_i to x_j is exactly $\ell' * d + 1$, (ii) a point is placed at each of the co-ordinates of the form $(d * i, d * j)$, where i and j are integers (shown using small squares in Figure 3.1(c)), (iii) one of the two segments adjacent to the point x_i or x_j contains exactly d newly added points and other segments on the path from x_i to x_j have $d - 1$ points (shown using small circles in Figure 3.1(c)), and (iv) only consecutive points on the path $x_i \rightsquigarrow x_j$ are within unit distance apart.

Now, we construct a UDG $G'' = (V'', E'')$, where $V'' = X \cup Y$, and $E'' = \{p_i p_j \mid p_i, p_j \in V'' \text{ and } d(p_i, p_j) \leq 1\}$. Here $|V''| = |X| + |Y| = n + \ell d$, and $|E''| = \ell d + m$, where m is the number of edges in G. Thus, G'' can be constructed in polynomial time. We will use the term *d-grid* for a grid whose each cell is of size $d \times d$.

The notion of points and vertices of G'' are used interchangeably in the rest of the chapter. Unless otherwise specified, the term distance refers to graph-distance.

Lemma 3.1.7. *Each DdIS of G'' contains at most ℓ points from Y.*

Proof. For each segment in the d-grid used to draw G', the number of points of Y appearing on it is d or $d - 1$. Thus, each segment may contain at most one point from Y in the DdIS of G''. In particular, if two end-points of a segment η of the d-grid (that are vertices of G'') are chosen in DdIS, then no point of Y lying on η will be chosen. Now, the result follows from the fact that ℓ many segments of the d-grid are used to draw G'. □

Lemma 3.1.8. *G has a DdIS of cardinality at least k if and only if G'' has a DdIS of cardinality at least $k + \ell$.*

Proof. (**Necessity**) Let G has a DdIS I of size at least k. Let $X' = \{x_i \in X \mid v_i \in I\}$. Let $G_{i,\alpha}$ denote a spanning tree of G with the set of vertices $V_{i,\alpha} = \{v_j \in V(G) \mid d_G(v_i, v_j) \leq \alpha\}$. For each $v_i \in I$ start traversing from x_i in G''. Let $Y_i = \{y_\theta \in Y \mid$

$d_{G''}(x_i, y_\theta) = d * \theta,\ \mathit{for\ all}\ \theta = 1, 2, \ldots, \ell'\}$, where ℓ' is the number of segments between x_i and x_j, where x_j corresponds to $v_j \in V_{i,\lfloor \frac{d}{2} \rfloor}$. Let $Y' = \bigcup_{x_i \in X'} Y_i$. The set $X' \cup Y'$ is a DdIS in G''. Observe that there are some segments (corresponding to the edges which are not part of any $G_{i,\lfloor \frac{d}{2} \rfloor}$) that have not been traversed in the above process. Now, we consider every such segment and choose the $\lceil \frac{d}{2} \rceil$-th point on it. Let Y'' be the set of chosen points. Needless to say, Y'' is also a DdIS of G''.

By the way, we obtained the sets Y' and Y'', there exist no pair of points $y_\alpha \in Y'$ and $y_\beta \in Y''$ such that $d_{G''}(y_\alpha, y_\beta) < d$. On the contrary, suppose $d_{G''}(y_\alpha, y_\beta) < d$. Implies, y_α and y_β are from two segments, each having one, incident at some $x_j \in X \setminus X'$, where x_j corresponds to a leaf $v_j \in G_{i,\lfloor \frac{d}{2} \rfloor}$. Note that $d_{G''}(y_\alpha, x_j) \geq \lfloor \frac{d}{2} \rfloor$ and $d_{G''}(x_j, y_\beta) \geq \lceil \frac{d}{2} \rceil$. Implies, $d_{G''}(y_\alpha, y_\beta) \geq d$, arrived at a contradiction. Let $I' = X' \cup Y' \cup Y''$. As per our selection method each segment contributes one point in $Y' \cup Y''$. Thus, $|I'| \geq k + \ell$ since $|X'| \geq k$ and $|Y' \cup Y''| = \ell$.

(**Sufficiency**) Let G'' has a DdIS I' of cardinality at least $k + \ell$ and $I = \{v_i \in V \mid x_i \in I' \cap X\}$. Observe that $|I' \cap Y| \leq \ell$ (due to Lemma 3.1.7); so $|I| \geq k$. We shall show that, by suitably modifying I (i.e., by removing or changing some of the vertices in I), we get at least k points from X such that the set of corresponding vertices in G is a DdIS of G. Consider a pair of vertices $v_i, v_j \in I$ such that $d_G(v_i, v_j) = d' < d$ in G (if there is no such pair, then I is a DdIS of G with $|I| \geq k$). Let $x_i, x_j \in I' \cap X$ be the vertices in G'' corresponding to $v_i, v_j \in I$, respectively. Also, let $\hat{\ell}$ be the number of segments on the path $x_i \rightsquigarrow x_j$ corresponding to the shortest path $v_i \rightsquigarrow v_j$. As each segment can contribute at most one point (from Y) in any solution, I' can contain at most $\hat{\ell} + 1$ points (including x_i and x_j) from the path $x_i \rightsquigarrow x_j$.

As per the construction of G'', the distance between x_i and x_j is $\hat{\ell} * d + 1$ in G''. We update the solution along the path $x_i \rightsquigarrow x_j$ as follows: delete x_j and other points of the path from I'. Start traversing the path from x_i and add every $(d * \theta)-$th point to I', where $1 \leq \theta \leq \hat{\ell}$. The last point chosen is the point which is d' distance away from x_j. The number of points in I' on the path $x_i \rightsquigarrow x_j$ is $\hat{\ell} + 1$. Thus, I' a new feasible solution in G'' whose size is at least as that of the previous solution. Observe that, the points in I' that are on the segments outside the path $x_i \rightsquigarrow x_j$ will not be affected by the newly chosen points, and $|I'| \geq k + \ell$.

We repeat the same for all pair of points in I for which the shortest path distance in G is less than d. Therefore, $|I| \geq k$ (from Lemma 3.1.7) and I is a distance-d independent set in G. □

Theorem 3.1.9. *The D-GMDdIS problem belongs to the class NP-complete.*

Proof. Follows from Lemma 3.1.1 and Lemma 3.1.8. □

3.2 Approximation Algorithm for GMDdIS Problem

In this section, we discuss a simple 4-factor approximation algorithm for the GMDdIS problem, for a fixed integer $d \geq 3$. Let $\mathcal{R}$ be the rectangular region containing the point set P (disk centers). From now on we deal with the point set P rather than the UDG G defined on P. We partition $\mathcal{R}$ into disjoint horizontal strips $H_1, H_2, \ldots, H_\nu$, each of width d (H_ν may be of width less than d). The basic idea behind our algorithm is as follows:

(i) Compute a feasible solution for each non-empty strip H_i ($1 \leq i \leq \nu$) independently as stated below:

We split the horizontal strip into squares of size $d \times d$ (see Figure 3.2). In each square, we compute an optimum solution of the DdIS problem defined by the points set inside that square. We consider all odd-numbered squares and compute the union S^i_{odd} of optimum solutions of these squares. Similarly, the union S^i_{even} of optimum solutions of all even-numbered squares are also computed. Each of these is a feasible solution of DdIS problem in H_i as the minimum distance between each pair of considered squares is at least d. We

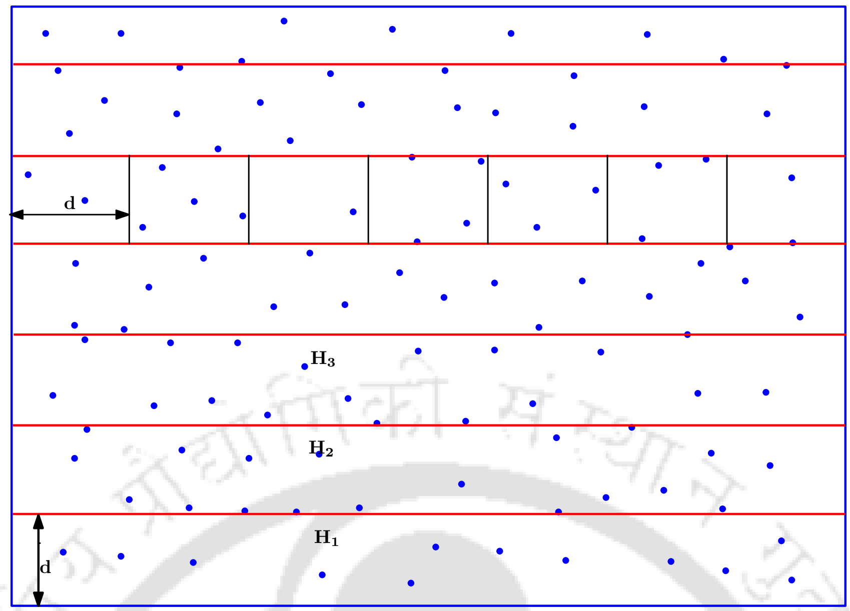


Figure 3.2: A horizontal and vertical partition of the strips of width d

choose $S^i = S^i_{even}$ or S^i_{odd} such that $|S^i| = \max(|S^i_{even}|, |S^i_{odd}|)$ as the desired feasible solution for the strip H_i.

(ii) Compute S_{even} and S_{odd}, which are the union of the solutions of even and odd strips respectively, and

(iii) Report $S^* = S_{even}$ or S_{odd} such that $|S^*| = \max(|S_{even}|, |S_{odd}|)$ as a solution to the DdIS problem.

Note that, the solution obtained in the above process is a feasible solution for the entire problem.

Lemma 3.2.1. *If OPT is an optimum solution for the GMDdIS problem, then* $\max(|S_{even}|, |S_{odd}|) \geq \frac{1}{4}|OPT|$.

Proof. Let us denote by OPT^i an optimum solution of the non-empty strip H_i. Since any two even (resp. odd) numbered strips, say H_i and H_j, are at least d distance apart, the feasible solutions computed in any method for H_i and H_j are independent[1]. Thus,

[1]by independent we mean for any $p_i \in H_i \cap P$ and $p_j \in H_j \cap P$, p_i and p_j are distance-d independent and also, $OPT^i \cap OPT^j = \emptyset$

both $OPT_{even} = \bigcup_{i \texttt{ is even}} OPT^i$ and $OPT_{odd} = \bigcup_{i \texttt{ is odd}} OPT^i$ are feasible solutions for the given DdIS problem.

Note that $|OPT| \leq |OPT_{even}| + |OPT_{odd}| \leq 2|OPT_*|$, where $OPT_* = OPT_{even}$ if $|OPT_{even}| > |OPT_{odd}|$; otherwise $OPT_* = OPT_{odd}$.

Also, note that we have not computed OPT^i for the strip H_i. Instead, we have computed S^i_{even} and S^i_{odd} by splitting the strip H_i into $d \times d$ squares, and accumulating the optimum solutions of even and odd numbered squares separately. By the same argument as stated above, we have $|OPT^i| \leq 2|S^{i_*}|$, where $S^{i_*} = S^i_{even}$ if $|S^i_{even}| \geq |S^i_{odd}|$; otherwise $S^{i_*} = S^i_{odd}$.

Combining both the inequalities, we have $|OPT| \leq 4 \max(|S_{even}|, |S_{odd}|)$, i.e., max $(|S_{even}|, |S_{odd}|) \geq \frac{1}{4}|OPT|$. □

3.2.1 Optimal solution of a $d \times d$ square problem

Let $\mathcal{Q} \subseteq P$ be the set of points inside a $d \times d$ square χ, and G_χ be the UDG defined on $\mathcal{Q}$. Let $C_1, C_2, \ldots, C_l$ be the connected components of G_χ. Without loss of generality we assume that any two components in G_χ are at least d distance apart[2] in G.

Lemma 3.2.2. *The maximum number of different connected components in G_χ is $O(d^2)$.*

Proof. Partition χ into $O(d^2)$ cells, each of size $\frac{\sqrt{3}}{2} \times \frac{1}{2}$. The result follows from the fact that the points lying inside each cell are mutually connected. □

To have the worst-case size of a DdIS in G_χ, we need to have an idea about the worst-case size of a DdIS in a connected component in G_χ.

Lemma 3.2.3. *Let C be any connected component of G_χ. The number of mutually distance-d independent points in C is bounded by $O(d)$.*

Proof. Consider the square region χ' of size $3d \times 3d$ whose each side is d distance away from the corresponding side of χ. Let $Q' \subseteq P$ be the subset of points in χ'. Partition χ'

[2]if there are two components having distance less than d in G, then we can view them as a single component

into cells of size $\frac{1}{2\sqrt{2}} \times \frac{1}{2\sqrt{2}}$. Thus, the number of cells in χ' is bounded by $O(d^2)$, and in each cell the unit disks centered at the points inside that cell are mutually connected. Let a pair of points $p_i, p_j \in C$ which are distance-d independent. The shortest path $p_i \rightsquigarrow p_j$ between p_i and p_j entirely lies inside χ'. If there is another point $p_k \in C$ which is distance-d independent with both p_i and p_j, then p_k is at least distance $\frac{d}{2}$ away from each point on the path $p_i \rightsquigarrow p_j$. Therefore, the path from p_k to any point on the path $p_i \rightsquigarrow p_j$ occupies at least $O(d)$ cells, and none of the points from these cells are distance-d independent to all the points p_i, p_j, p_k. Thus, the addition of each point in the set of mutually distance-d independent points in χ prohibits points in $O(d)$ cells to belong in that set, and hence the lemma follows. □

Lemma 3.2.4. *An optimal (i.e., maximum size) DdIS in χ can be computed in $d^2 n^{O(d)}$ time.*

Proof. We first construct a weighted complete graph $G' = (V', E')$ where V' corresponds to the points in Q'. For each edge $v_i v_j \in E'$, the weight $w(v_i v_j) = 1$ if $d(p_i, p_j) \leq 1$; otherwise $w(v_i v_j) = \infty$. Next, we compute the all pair shortest paths between every pair of vertices in G', and store them in a matrix $\mathcal{M}$.

By definition, the intersection of distance-d independent sets of any two components is empty. Thus, a DdIS of maximum size in G_χ can be computed by considering all components of the UDG G_χ, and computing the union of the DdIS of maximum sizes of those components. We consider each component of G_χ separately. For each component C, we consider all possible tuples of size at most $O(d)$ (due to Lemma 3.2.3) and for each tuple, we check whether they form a DdIS or not by consulting the matrix $\mathcal{M}$ in $O(d^2)$ time. Thus, a maximum size DdIS in C can be computed in $O(d^2|C|^{O(d)})$ time and the total time for computing a maximum size DdIS in G_χ is $O(d^2 \sum_{C \in G_\chi} |C|^{O(d)}) = d^2 n_\chi^{O(d)}$, where $n_\chi = \sum_{C \in G_\chi} |C|$, the number of vertices in G_χ. □

Theorem 3.2.5. *Given a set P of n points in the plane, a DdIS of size at least $\frac{1}{4}|OPT|$ can be computed in $d^2 n^{O(d)}$ time, where $|OPT|$ is the maximum cardinality of a GMDdIS.*

Proof. Follows from Lemmata 3.2.1, 3.2.2, 3.2.3 and 3.2.4. □

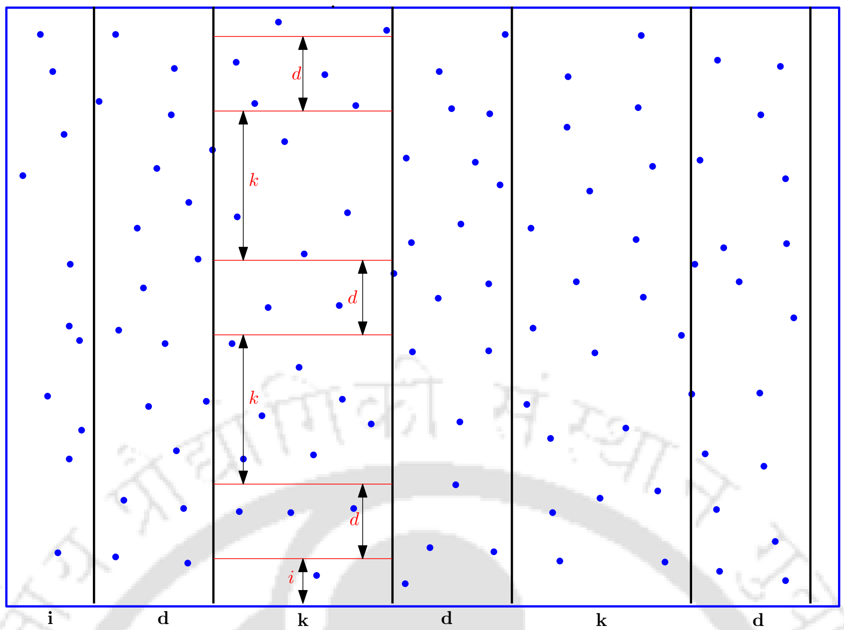


Figure 3.3: Horizontal and vertical partition of the strips

3.3 Approximation Scheme for GMDdIS Problem

In this section, using the shifting strategy [55], we propose a polynomial time approximation scheme (PTAS) for the GMDdIS problem, for a given fixed integer $d \geq 3$. Let $\mathcal{R}$ be an axis parallel rectangular region containing the point set P (i.e., centers of the disks of the given UDG). We use two-level nested shifting strategy. The first level executes k iterations, where $k \gg d$. The i-th iteration ($1 \leq i \leq k$) of the first level is as follows:

- Assuming $\mathcal{R}$ is left-open, partition $\mathcal{R}$ into vertical strips such that (a) first strip is of width i, (b) every even strip is of width d, and (c) every odd strip, except the first strip, is of width k (see Figure 3.3).
- Without loss of generality, assume that the points lying on the left boundary of a strip belong to the adjacent strip to its left (i.e., every strip is left open and right closed).
- Compute some desired feasible solutions for the odd strips (of width k). These solutions can be merged to solve the entire problem since these odd-numbered

strips are distance-d apart.

The second level of the nested shifting strategy is used to find a solution for an iteration in the first level. We consider each non-empty odd strip separately and execute k iterations. In the i-th iteration, we partition it horizontally as in the first level (mentioned in the first bullet above). We get a solution of a strip by solving each $k \times k$ square in that strip optimally. The union of the solutions of all the odd-numbered squares/rectangles in that strip is the desired solution of that vertical strip of the first level. Finally, we take the union of the solutions of all the odd vertical strips to compute the solution of that iteration of the first level. Thus, we have the solutions of all the iterations of the first level. We report the one having the maximum cardinality as the solution of the given DdIS problem. Compute a matrix $\mathcal{M}$ containing the cost of all pair shortest paths in a complete graph defined with the points in P where the edge costs are as defined in Section 3.2.1. The method of computing an optimum solution inside a $k \times k$ square is described below.

3.3.1 Computing an optimum solution in a $k \times k$ square

We apply a divide-and-conquer strategy to compute an optimum solution of the GMDdIS problem defined on a set of points $\mathcal{Q} \subseteq P$ inside a square χ of size $k \times k$. We partition χ into four sub-squares, each of size $\frac{k}{2} \times \frac{k}{2}$, using a horizontal line ℓ_h and a vertical lines ℓ_v (see Figure 3.4). Let $\mathcal{Q}_1 \subseteq \mathcal{Q}$ be the subset of points in χ which are at most d distance away from ℓ_h and/or ℓ_v. Let $\mathcal{Q}_2$ be a maximum cardinality subset of $\mathcal{Q}_1$ such that all the points in $\mathcal{Q}_2$ are pair wise distance-d independent in P.

Lemma 3.3.1. *$|\mathcal{Q}_2| \leq O(k)$.*

Proof. The proof follows from a similar combinatorial argument discussed in the proof of Lemma 3.2.3. □

We apply the divide-and-conquer strategy on χ as follows:

Step 1: Choose all possible subsets of points of sizes at most $O(k)$ in $\mathcal{Q}_1$.

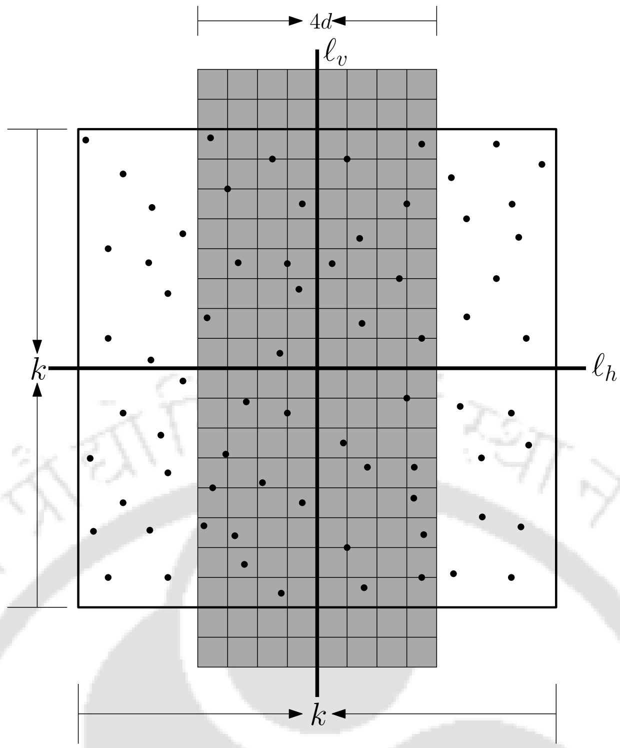


Figure 3.4: A strip of width $4d$ around the vertical line ℓ_v (shown in dotted lines)

Step 2: For each subset, do the following:

- Check whether they are mutually distance-d independent by consulting the table $\mathcal{M}$. If so, then they form $\mathcal{Q}_2$.
- Consult the matrix $\mathcal{M}$ to delete the points in χ which are at most distance $d-1$ away from each member in $\mathcal{Q}_2$.
- Recursively solve the four independent sub-problems defined by the points of $\mathcal{Q} \setminus \mathcal{Q}_1$ in the four quadrants χ_1, χ_2, χ_3, χ_4 defined by ℓ_h and ℓ_v.
- Return $\mathcal{Q}_2 = \mathcal{Q}_2 \cup (\bigcup_{i=1}^{4} \mathcal{Q}_2^i)$, where $\mathcal{Q}_2^i$ is the solution of the sub-problem on the points of χ_i.
- Retain the solution for the present subset if it is better than the solutions produced by earlier choices of $\mathcal{Q}_2$.

Lemma 3.3.2. *The solution produced for the cell χ (of size $k \times k$) in the aforesaid* *process is optimum, and the time complexity of the proposed algorithm is $k^2 m^{O(k)}$, where $m = |\mathcal{Q}|$.*

Proof. Let OPT_χ be an optimal solution for the points lying in χ. Note that our process checks all combinations of points of size $|OPT_\chi|$. Thus, the combination of points in OPT_χ must appear at some stage in the process.

If $T(m,k)$ denote the time complexity of computing the distance-d independent set in χ, then $T(m,k) = 4 \times T(m, \frac{k}{2}) \times m^{O(k)} + O(k^2)$, which is $k^2 \times m^{O(k)}$ in the worst case. □

Using the analysis of [55], we have the following result.

Theorem 3.3.3. *Given a set P of n points (centers of the unit disks) in the plane and an integer $k > 1$, the proposed scheme produces a DdIS of size at least $\frac{1}{(1+\frac{1}{k})^2}|OPT|$ in $k^2 n^{O(k)}$ time, where $|OPT|$ is the maximum cardinality of a GMDdIS.*

3.4 The Hardness Result of GMDdDS Problem

In this section, we discuss the hardness result of the GMDdDS problem by proving the decision version of the GMDdDS problem belongs to the class NP-complete. The decision version of the GMDdDS problem, denoted by D-GMDdDS, for a fixed integer $d \geq 2$, is defined as follows:

> *Given a unit disk graph $G = (V, E)$ defined on a point set P and a positive integer $k \leq |V|$, does there exist a distance-d dominating set of size at most k in G?*

Lemma 3.4.1. *D-GMDdDS problem belongs to the class NP.*

Proof. For a given set of vertices, we can verify whether all the vertices of the input graph are distance-d dominated or not in polynomial-time using Floyd-Warshall's all-pair shortest path algorithm [28]. Hence D-GMDdDS $\in$ NP. □

Now, to prove the problem belongs to the NP-hard class, we do a polynomial-time reduction from a known NP-hard problem, the *vertex cover* problem [43] defined in planar graphs with maximum degree 3, to it. The decision version of the vertex cover problem in planar graphs with maximum degree 3, denoted by D-VC_p, is defined as follows:

Given an undirected planar graph G with maximum degree 3 and a positive integer k, does there exist a vertex cover D of G such that $|D| \leq k$?

Corollary 3.4.2. *A planar graph $G = (V, E)$ with maximum degree 3 can be embedded on a plane having grid cell of size $2d \times 2d$, so that its vertices lie at points of the form $(i * 2d, j * 2d)$ and its edges are drawn using a sequence of consecutive line segments drawn on the vertical lines of the form $x = i * 2d$ and/or horizontal lines of the form $y = j * 2d$, for some integers i and j* (see Figure 3.5).

Proof. Follows from Lemma 3.1.4. □

Lemma 3.4.3. *Let $G = (V, E)$ be an instance of* D-VC_p *with maximum degree 3. An instance $G' = (V', E')$ of D-GMDdDS can be constructed from G in polynomial-time.*

Proof. Embed the instance G of D-VC_p on the plane as discussed in Corollary 3.4.2, using one of the algorithms in [56, 59]. An edge in the embedding is a sequence of connected line segment(s) of length $2d$ units each. Let ℓ be the total number of line segments in the embedding. We add points $P = \{p_1, p_2, \ldots, p_n\}$ corresponding to the vertices $V = \{v_1, v_2, \ldots, v_n\}$ in the embedding. To make G' a UDG we introduce a set Q of extra points on the segments that are used to draw the edges of G'. Thus, the set of points in P together with Q form a UDG G'. Let p_ip_j be an edge in G' corresponding to the edge v_iv_j in G and has ℓ' segments in the embedding.

Case 1: If $\ell' = 1$, then we add $3d$ points $p_{ij}^1, p_{ij}^2, \ldots, p_{ij}^{3d}$ on the segment such that the Euclidean distance of p_i to p_{ij}^1 and p_{ij}^{3d} to p_j is 0.72 and the Euclidean distance between p_{ij}^t and p_{ij}^{t+1} is $\frac{2d-1.44}{3d-1} > 0.5$ for $t = 1, 2, 3, \ldots, 3d-1$. Therefore, the length of the path from p_i to p_j is exactly $3d + 1$ (for $d = 2$, see the edge p_1p_4 in Figure 3.5(c)).

Case 2: If $\ell' > 1$, then we consider all joint points of each pair of consecutive segments other than the points of P and add a point for each joint point to the set Q (see the

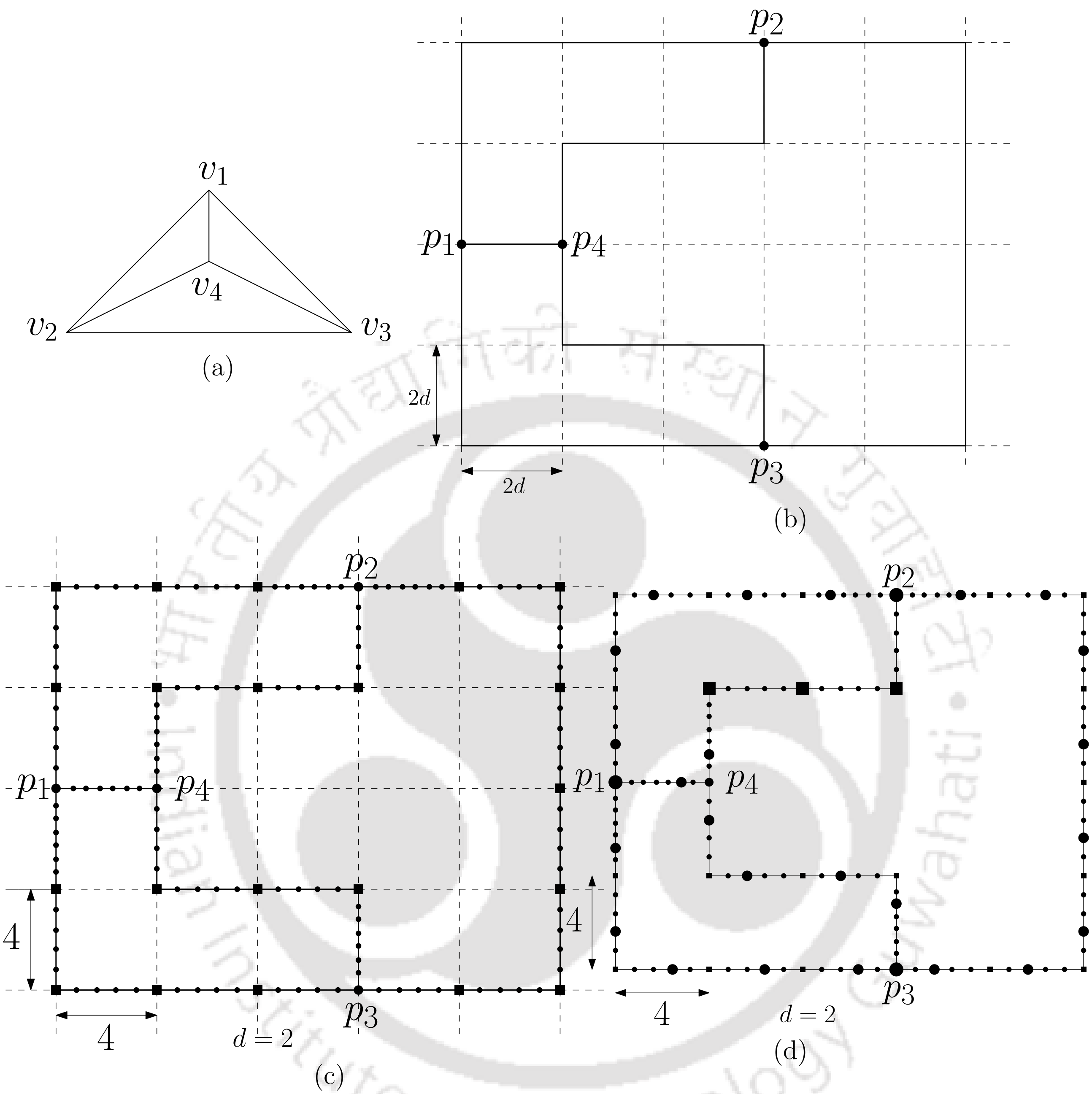


Figure 3.5: (a) A planar graph G of maximum degree 3, (b) its embedding G' on a grid of cell size 4×4, (c) adding of extra points to G', and (d) the obtained UDG G'.

square points in the edge p_1p_3 in Figure 3.5(c)). Then, we add $3d$ points as in Case 1 in one of the two segments for which one end is associated either with p_i or p_j; and for each other segment(s) we also add $2d$ points $p_{ij}^1, p_{ij}^2, \ldots, p_{ij}^{2d}$ such that the Euclidean distance to the points p_{ij}^1 and p_{ij}^{2d} from the end points of the segment(s) is 0.75 and the

Euclidean distance between p_{ij}^t and p_{ij}^{t+1} is $\frac{2d-1.5}{2d-1} > 0.5$ for $t = 1, 2, 3, \ldots, 2d-1$ in such a way that after adding the extra points, the path length of one segment is $3d+1$ and the path length of all other segments is $2d+1$ (see the edge p_2p_3 in Figure 3.5(c)). Note that in both the cases, only consecutive pair of points on the path $p_i \rightsquigarrow p_j$ are within unit distance apart. Let Q be the set of all extra points added in either of the cases for each segment.

Observe that $G' = (V', E')$ is a UDG, where $V' = P \cup Q$, and $E' = \{p_ip_j \mid p_i, p_j \in V' \text{ and } d(p_i, p_j) \leq 1\}$. Here $|V'| = |P| + |Q| \leq n + 3\ell d$, and $|E'| \leq 3\ell d + m$, where m is the number of edges in G and ℓ is bounded by n. Thus, G' can be constructed in polynomial-time. □

Lemma 3.4.4. *G has a vertex cover of size at most k if and only if G' has a distance-d dominating set of size at most $k + \ell$.*

Proof. (**Necessity**) Let D be a vertex cover for the graph G such that $|D| \leq k$. Let $\mathcal{S}$ be the collection of points from P in G' corresponding to the vertices of D in G, i.e., $\mathcal{S} = \{p_i \in P \mid v_i \in D\}$. Note that $|\mathcal{S}| = |D|$. We choose one point from each segment in such a way that the selected points along with $\mathcal{S}$ form a distance-d dominating set of G' and the size of $\mathcal{S}$ is at most $k + \ell$.

Note that, every edge in G has at least one of its end vertices in D (D is a vertex cover in G). For each edge v_iv_j in G, start traversing from the corresponding vertex p_i in G' (if $v_i \in D$ or from p_j, if $v_j \in D$) in the embedding and select each $(2d+1)$-st vertex in $\mathcal{S}$ encountered from p_i to p_j in the traversal (see p_2p_3 in Figure 3.5(d). The big vertices are part of $\mathcal{S}$ while traversing from p_2). Observe that, $\mathcal{S}$ is a distance-d dominating set in G' having $|\mathcal{S}| \leq k + \ell$ as we have chosen one vertex from each segment in the embedding and the way we have chosen $\mathcal{S}$, for any $p_i \in V'$ there is always exist at least one point $p_j \in \mathcal{S}$ such that $d(p_i, p_j) \leq d$.

(**Sufficiency**) Let $\mathcal{S} \subseteq V'$ be a DdDS of size at most $k + \ell$ in G'. We need to prove that G has a vertex cover of size at most k. Let $D = \{v_i \in V \mid p_i \in \mathcal{S} \cap P\}$. Observe that $|D| \leq k$ as the length of each segment in G' is at least $2d+1$ there must be at least

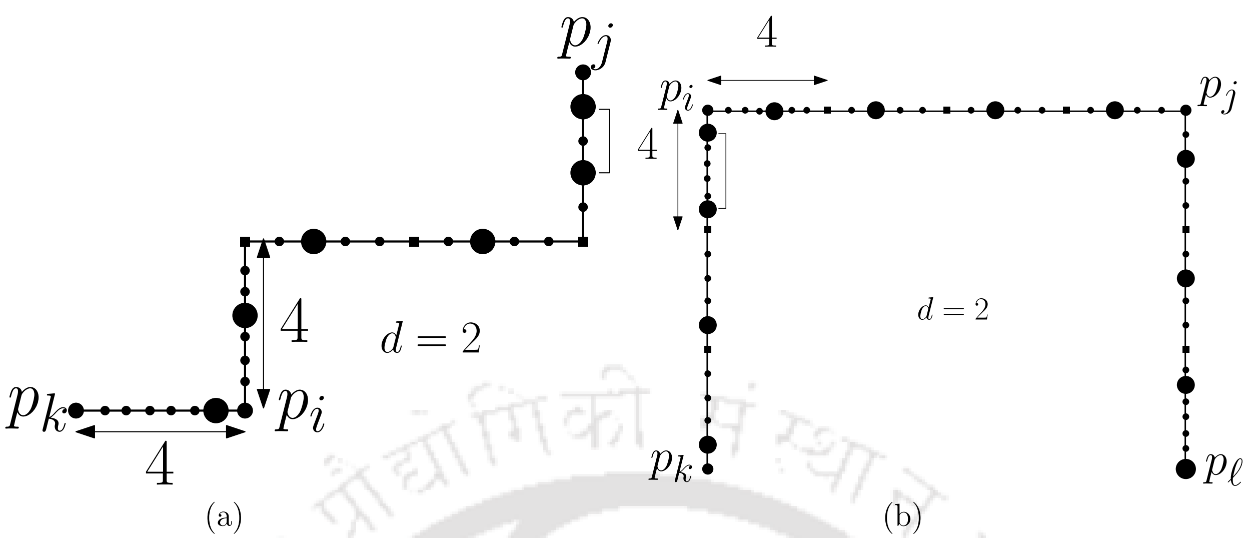


Figure 3.6: (a) p_j is only connected with p_i, and (b) p_i connected with p_k and p_j connected with p_ℓ.

one point from each segment chosen in $\mathcal{S}$. It remains to prove that D is a vertex cover of G. If any edge $v_i v_j$ in G has none of its end vertices in D, then consider the points p_i and p_j corresponding to v_i and v_j respectively.

Case (i): Degree of $v_j = 1$ in G. Let ℓ' be the number of segments on the path from $p_i \rightsquigarrow p_j$ in G'. Observe that from the path $p_i \rightsquigarrow p_j$ there are at least $\ell' + 1$ vertices in $\mathcal{S}$ (see Figure 5.3(a) for example). In this case, we delete one point from the segment containing two points in $\mathcal{S}$ and introduce p_i in $\mathcal{S}$. A similar argument works even if degree of $v_i = 1$ in G.

Case (ii): Degrees of both v_i and v_j are greater than 1 in G. Both p_i and p_j are connected with some points p_k and p_ℓ respectively in G', then either the chain of segments (say ℓ') in the path $p_i \rightsquigarrow p_j$ in G' has at least $\ell' + 1$ vertices in $\mathcal{S}$ (see Case (i)) or the chain of segments (say ℓ') in the path $p_i \rightsquigarrow p_k$ or ($p_j \rightsquigarrow p_\ell$) in G' has at least $\ell' + 1$ vertices in $\mathcal{S}$ (see Figure 5.3(b) for example). In this case, we choose the segment having two points in $\mathcal{S}$ and remove one point of the segment from $\mathcal{S}$ and introduce p_j in $\mathcal{S}$ if $p_k \in \mathcal{S}$ otherwise introduce p_i in $\mathcal{S}$. Update D and repeat the process till every edge has at least one of its end vertices in D. Note that, in both the cases, we delete at most one point from such segments having two of its points in the solution and there does not exist a segment in G' having none of its points in $\mathcal{S}$, which leads the proof that D is a vertex cover in G with $|D| \leq k$. □

Theorem 3.4.5. *The D-GMDdDS problem belongs to the class NP-complete.*

Proof. Follows from Lemma 3.4.1 and Lemma 3.4.4. □

3.5 Approximation Algorithm for GMDdDS Problem

In this section, we propose a 4-factor approximation algorithm for the minimum distance-d dominating set problem. Let $\mathcal{R}$ be the axis parallel smallest rectangular region containing the point set P (disk centers). We partition $\mathcal{R}$ into squares having side length $\frac{3}{\sqrt{2}}d \times \frac{3}{\sqrt{2}}d$ (see Figure 3.7(a)). The basic idea behind the propose algorithm is as follows:

- color the partitioning squares with 4-colors such that the distance between two same colored squares are more than $2d$ (see Figure 3.7(a)).
- find an optimal solution of each square (see Subsection 3.5.1).
- let OPT_i denotes the union of optimal solutions generated by our algorithm for the squares having color i, for $i = 1, 2, 3, 4$.
- Let OPT be an MDdDS of the UDG defined on P. Therefore, $|OPT_i| \leq |OPT|$. Thus, $\sum_i |OPT_i| \leq 4 * |OPT|$.

3.5.1 Computing a MDdDS in a $\frac{3}{\sqrt{2}}d \times \frac{3}{\sqrt{2}}d$ square

Let χ be a single $\frac{3}{\sqrt{2}}d \times \frac{3}{\sqrt{2}}d$ cell. Also, let $P_\chi \subseteq P$ be the set of points inside χ, and G_χ be the UDG defined on P_χ. Let $C_1, C_2, \ldots, C_\ell$ be the different connected components of G_χ with the constraint that each C_i $(1 \leq i \leq \ell)$ are d-distance apart from each other in G (i.e., the UDG corresponding to points set P). If the distance between any two components is less than d, then combine these two components as a single component.

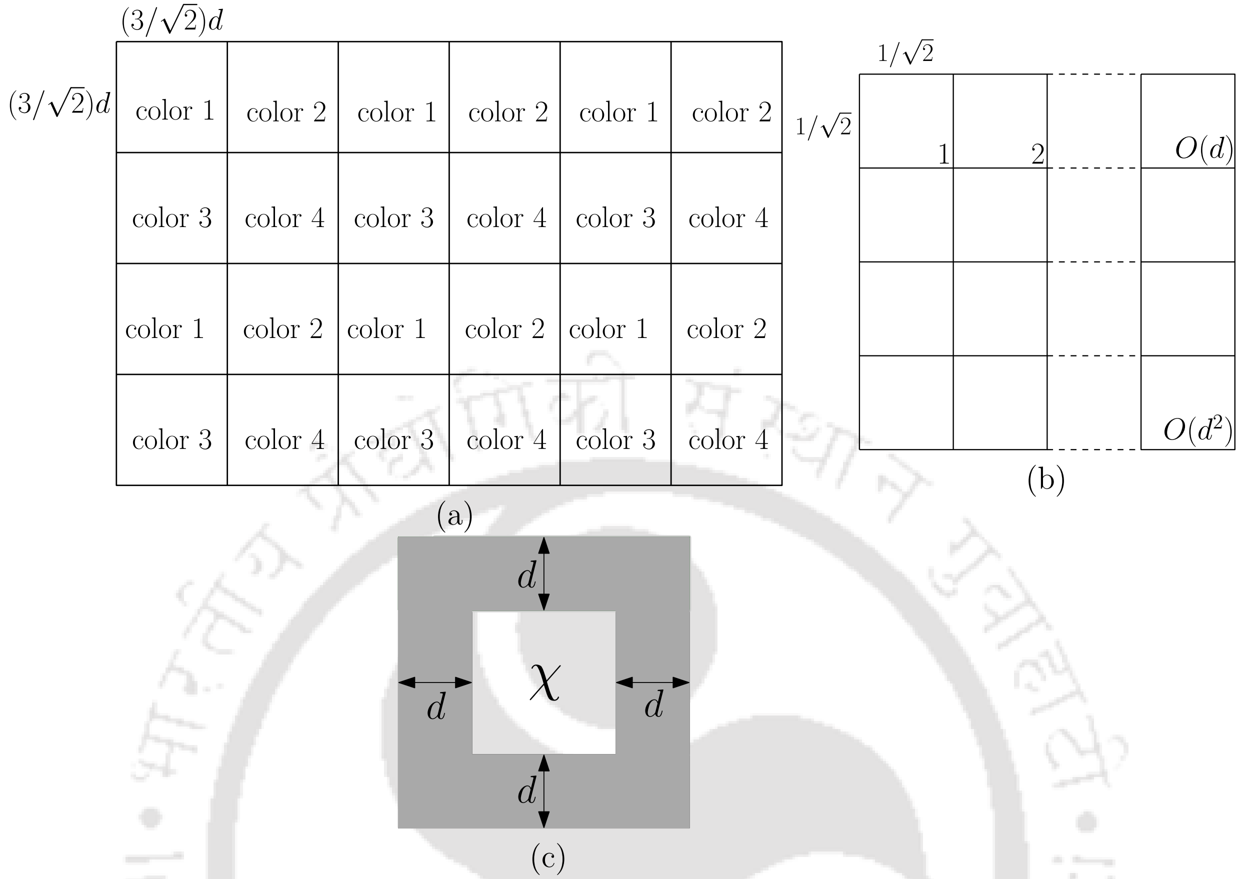


Figure 3.7: (a) partition of R into smaller cells of size $\frac{3}{\sqrt{2}}d \times \frac{3}{\sqrt{2}}d$, (b) one cell partitioned into $O(d^2)$ sub-cells, and (c) one cell surrounded with d width region.

Lemma 3.5.1. *The number of different connected components in G_χ is bounded by $O(d^2)$.*

Proof. Partition χ into sub-cells of size $\frac{1}{\sqrt{2}} \times \frac{1}{\sqrt{2}}$ (see Figure 3.7(b)). Therefore, the total number of sub-cells is $O(d^2)$. Every pair of points within a sub-cell are connected as they are at most unit distance apart. Therefore, the points lying inside each sub-cell are in the same connected component. □

Lemma 3.5.2. *The size of an MDdDS in any connected component $C \in \{C_1, C_2, \ldots, C_\ell\}$ of G_χ is bounded by $O(d)$.*

Proof. The proof follows from Lemma 3.2.3, with the fact that the minimum distance-d dominating set in any graph is bounded by the maximum cardinality distance-d independent set of the same graph. The same result holds for any sub-graphs also. □

Lemma 3.5.3. *The time complexity to compute an MDdDS in χ is $d^2 n^{O(d)}$.*

Proof. Consider a d-width region around the cell χ as χ' (see Figure 3.7(c)), having point set $P_{\chi'} \subseteq P$. Let $G_{P_{\chi'}}$ be a weighted complete graph with the vertex set corresponding to point set $P_{\chi'}$ and edge costs are as defined in Subsection 3.2.1. Apply all-pairs shortest path algorithm [28] on graph $G_{P_{\chi'}}$ and store the result in a matrix $\mathcal{M}$.

Observe that, for computing an MDdDS in graph G_χ, we need to compute an MDdDS in each and every component $C_i \in G_\chi$. As per the definition of components, all the components are d-distance apart from each other. So, taking the union of the computed solutions of each component leads to an MDdDS for the graph G_χ. For computing an optimum solution in a component C_i, we consider all possible tuples of size at most $O(d)$ (refer Lemma 3.5.2) and check whether the selected tuple is a feasible solution or not with the help of matrix $\mathcal{M}$ in $O(d^2)$ time. Thus, MDdDS can be computed in a single component C_i in $O(d^2|C_i|^{O(d)})$ time. Hence, computing an MDdDS in G_χ takes $O(d^2 \sum_{C_i \in G_\chi} |C_i|^{O(d)}) = d^2 n^{O(d)}$ time, where n is the number of vertices in G_χ. □

Theorem 3.5.4. *Given a set P of n points in the plane, a distance-d dominating set of size at most $4|OPT|$ can be computed in $d^2 n^{O(d)}$ time, where OPT is a minimum distance-d dominating set of the unit disk graph constructed on the point set P.*

Proof. Follows from Lemmata 3.5.1, 3.5.2, and 3.5.3. □

3.6 Approximation Scheme for GMDdDS Problem

In this section, using the technique of shifting strategy [55], we propose a polynomial-time approximation scheme (PTAS) for the MDdDS problem, for a given fixed integer d. Given a point set P (centers of the UDG) in an axis parallel rectangular region $\mathcal{R}$ and an integer $k \gg d$, we use two-level nested shifting strategy as follows:

- first, we apply shifting strategy in the horizontal direction. The i-th iteration $(1 \leq i \leq k)$ of the first level, partition $\mathcal{R}$ into horizontal strips such that the first strip is of width i, and the remaining strips are of width k. Note that the width of the last strip may be less than k.

- without loss of generality, assume that the points lying on the below boundary of a horizontal strip belong to its below adjacent strip.

- consider each non-empty horizontal strip H, and apply the second level of shifting strategy on the vertical direction.

- in the second level, the j-th iteration ($1 \leq j \leq k$) partition each non-empty horizontal strip H into square/rectangular cells of size (i) $j \times \ell$ for the first cell, and (ii) $k \times \ell$ for all other cells, where ℓ defines the width of the strip H ($\ell = i$ for the first strip and $\ell = k$ for all other strips).

We solve each $k \times k$ square (conceptually extend the smaller cells to $k \times k$ square) optimally, refer Subsection 3.6.1. For each horizontal strip H, we consider the union of the solutions of $k \times k$ squares in H to get a feasible solution for H. Finally, we take the union of the solutions of each non-empty horizontal strip to get a feasible solution of the problem in a single iteration. In the same process, we get the feasible solutions of all the iterations in the first level. We report the solution D having minimum cardinality among all the solutions generated in each iteration as the solution of the DdDS problem.

Now, we discuss the process to obtain an optimal solution in a $k \times k$ square. At the beginning, we compute a matrix $\mathcal{M}$ containing the cost of all pair shortest paths in a complete graph defined on the point set P, where the edge costs are defined as in Subsection 3.2.1.

3.6.1 Computing an optimum solution in a $k \times k$ square

We apply the same strategy as described in Subsection 3.3.1 on the point set $P_\chi \subseteq P$ inside a square χ of size $k \times k$. Let $P'_\chi \subseteq P_\chi$ denote the point set which are at most d distance away from ℓ_h and ℓ_v (the horizontal and vertical lines which divides χ into 4 squares). Let P''_χ be a minimum cardinality subset of P'_χ such that all the points in

P'_χ are distance-d dominated by the point set P''_χ. The following Lemma follows from Lemma 3.5.2.

Lemma 3.6.1. *$|P''_\chi|$ is bounded by $O(k)$.*

Lemma 3.6.2. *The optimum solution produced by our algorithm for each $k \times k$ square (χ) takes $k^2 n_\chi^{O(k)}$ time, where $n_\chi = |P_\chi|$ is the number of points inside χ.*

Proof. As our algorithm checks all combinations of points of size $|OPT_\chi|$, where OPT_χ is an optimal solution for χ, in some iteration the combination of points in OPT_χ must appear. Thus, time complexity result of the lemma follows from Lemma 3.3.2. □

Theorem 3.6.3. *Given a set P of n points in the plane and an integer $k \gg d$, a distance-d dominating set of size at most $(1 + \frac{1}{k})^2 \times |OPT|$ can be computed in $k^2 n^{O(k)}$ time, where OPT is the optimum solution.*

Proof. Let OPT be an MDdDS for the UDG G defined on the point set P, and $OPT' \subseteq OPT$ be the points chosen in OPT, which d-distance dominates the points outside the boundary of all the cells in an iteration (first level i-th iteration and second level j-th iteration). Let D^* be a solution obtained by our algorithm in an iteration.

Then, $|D^*| \leq |OPT| + |OPT'|$. For all the iterations of (i, j) $(1 \leq i, j \leq k)$, we have $\sum_{i=1}^{k}\sum_{j=1}^{k} |D^*| \leq k^2|OPT| + \sum_{i=1}^{k}\sum_{j=1}^{k} |OPT'|$. Since any point from a cell χ chosen in OPT can d-distance dominate points from no more than one horizontal strip (or vertical strip), and at most k times each horizontal (or vertical) boundary appears throughout the algorithm, we have $\sum_{i=1}^{k}\sum_{j=1}^{k} |OPT'| \leq k|OPT| + k|OPT|$. Thus, $\sum_{i=1}^{k}\sum_{j=1}^{k} |D^*| \leq k^2|OPT| + 2k|OPT| = (k^2 + 2k)|OPT|$. Therefore, $min \sum_{i=1}^{k}\sum_{j=1}^{k} |D^*| \leq (1 + \frac{1}{k})^2 \times |OPT|$.

The time complexity result follows from Lemma 3.6.2 along with the fact that, there are n points present in the plane. □

3.7 Conclusion

In this chapter, we studied the GMDdIS problem and the GMDdDS problem in unit disk graphs. We prove that, the decision version of both GMDdIS and GMDdDS problems belong to the class NP-complete. We proposed simple 4-factor approximation algorithms for both problems. We also proposed polynomial-time approximation schemes (PTAS) for both GMDdIS and GMDdDS problems.

Chapter 4

d-Distance m-Tuple (ℓ, r)-Dominating Set Problem in Graphs

The domination problem and its variants are the most studied problems in the literature for its wide range of applications. In this chapter, we study the *d-distance m-tuple* (ℓ, r)*-dominating set* ((d, m, ℓ, r) set) problem in general graphs. This problem gives a general framework for different variants of domination problem as well as for many other problems which are not studied in the literature. We define the minimum (d, m, ℓ, r) set problem formally as follows:

Definition 4.0.1. (Problem definition) *Given a simple undirected graph* $G = (V, E)$ *and positive integers* d, m, ℓ *and* r*, find a minimum size subset* $V' \subseteq V$ *of* G *satisfying the following two conditions: (i) for every* $v_i \in V$, $|N_G^d[v_i] \cap V'| \geq m$*, and (ii) for every* r *size subset* U *of* V, $|(\cup_{u \in U} N_G^d[u]) \cap V'| \geq \ell$*, where* $N_G^d[v_i]$ *is the d-distance neighborhood of a vertex* $v_i \in V$*, i.e.,* $N_G^d[v_i] = \{v_j \in V \mid d_G(v_i, v_j) \leq d$*, where* $d_G(v_i, v_j)$ *denotes the length of a shortest path between the vertices* v_i *and* v_j *in* $G\}$.

If $m \geq \ell$, then the second condition in the definition of (d, m, ℓ, r) set is redundant.

In the case of $m = \ell(= k$, say), the (d, m, ℓ, r) set is known as k-tuple dominating set in the literature. On the other hand, if $m = \ell$ then the value of $r > 1$ is irrelevant. Therefore, we assume $r = 1$ in case of $m = \ell$. From now onwards, we assume that $m \leq \ell$. If $d = 1, m = 2, \ell = 3, r = 2$ then (d, m, ℓ, r) set is known as a liar's dominating set in the literature. The objective of the d-distance m-tuple (ℓ, r)-domination problem is to find a minimum size d-distance m-tuple (ℓ, r) dominating set in a given graph G, and we call this problem as the *minimum (d, m, ℓ, r) set* problem. Here, our objective is to give a general framework such that variant problems of (i) dominating set, (ii) k-tuple dominating set, (iii) liar's dominating set, and (iv) many other problems depending on the values of d, m, ℓ, and r (which are not studied in the literature) come under one umbrella. In this chapter, we prove the hardness of d-distance m-tuple (ℓ, r)-domination problem and the inapproximability of the problem.

4.1 Hardness Results

4.1.1 Hardness Result of the 1-Distance m-Tuple (ℓ, r)-Domination Problem

In this section, we show that the decision version of the 1-distance m-tuple (ℓ, r)-domination problem in graphs belongs to the class NP-complete by reducing the *dominating set* (DS) problem to it, which is known to be in the class of NP-complete [43].

The definition of the decision version of both problems are as follows:

Definition 4.1.1. (Decision version of 1-distance m-tuple (ℓ, r)-domination problem) *Given a simple undirected graph $G = (V, E)$ with at least ℓ vertices and three positive integers m, r, and $k(\leq |V|)$, where $m \leq \ell$, does there exist a 1-distance m-tuple (ℓ, r)-dominating set of size at most k?*

Definition 4.1.2. (Decision version of the DS problem)

Given a simple undirected graph $G = (V, E)$ and a positive integer k, does there exist a dominating set D of G such that $|D| \leq k$?

Theorem 4.1.3. *The decision version of the 1-distance m-tuple (ℓ, r)-domination problem belongs to the class NP-complete.*

Proof. For any given set $L \subseteq V$ and a positive integer k, we can verify whether L is a 1-distance m-tuple (ℓ, r)-dominating set of size at most k or not in polynomial time by checking both the conditions of 1-distance m-tuple (ℓ, r)-dominating set. Therefore, 1-distance m-tuple (ℓ, r)-domination problem is in the class NP.

Now, we prove the hardness of the 1-distance m-tuple (ℓ, r)-domination problem by reducing the decision version of the DS problem, which is known to be NP-complete [43], to it. Let $< G = (V, E), k >$ be an arbitrary instance of the dominating set problem, where $G = (V, E)$ is an undirected graph with vertex set $V = \{v_1, v_2, \ldots, v_n\}$ and k is an integer. We construct an instance $< G' = (V', E'), m, \ell, r >$ of the decision version of 1-distance m-tuple (ℓ, r)-domination problem as follows:

$$V' = V^1 \cup V^2 \cup V^3, \text{where}$$
$$V^1 = \{v_1^1, v_2^1, \ldots, v_n^1\},$$
$$V^2 = \{v_1^2, v_2^2, \ldots v_{\ell-1}^2\},$$
$$V^3 = \{v_1^3, v_2^3, \ldots, v_r^3\}$$
$$E' = E^1 \cup E^2 \cup E^3 \cup E^4, \text{where}$$
$$E^1 = \{v_i^1 v_j^1 \mid v_i v_j \in E\},$$
$$E^2 = \{v_i^2 v_j^2 \mid 1 \leq i < j \leq \ell - 1\},$$
$$E^3 = \{v_i^1 v_j^2 \mid 1 \leq i \leq n, 1 \leq j \leq \ell - 1\},$$
$$E^4 = \{v_i^2 v_j^3 \mid 1 \leq i \leq \ell - 1, 1 \leq j \leq r\}$$

Since size of G' is polynomial, $G' = (V', E')$ can be constructed in polynomial-time and $|V'| = n + \ell + r - 1$, where $n = |V|$ and $\ell, r < n$. An illustration for the construction of G' from G is shown in Figure 4.1.

Claim 1: *G has a dominating set of size at most k if and only if G' has a 1-distance*

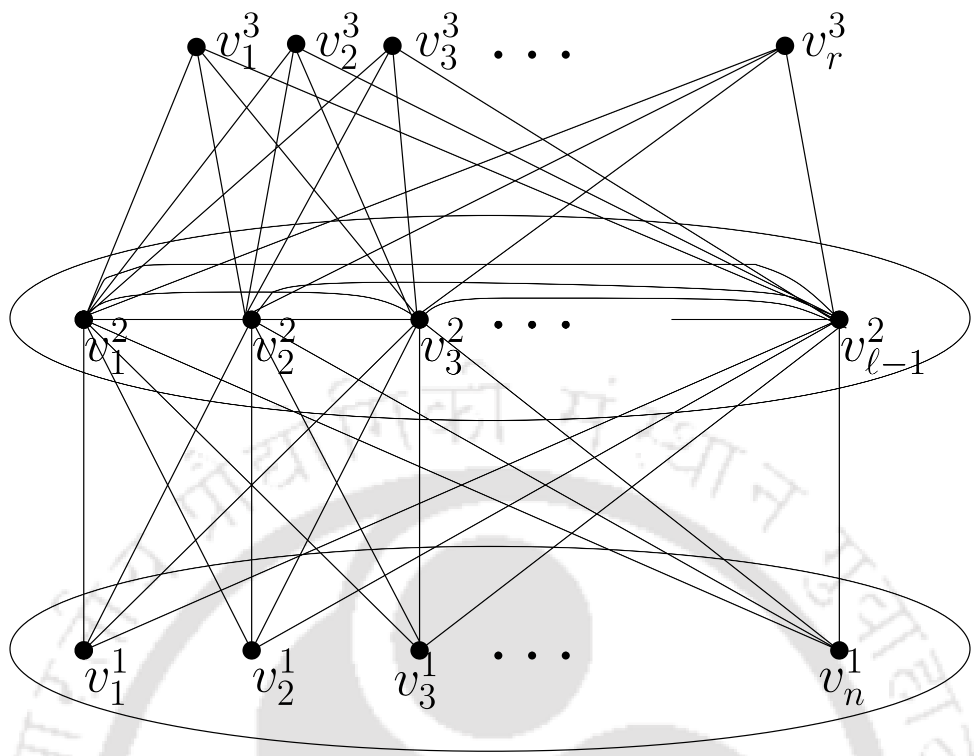


Figure 4.1: A graph $G' = (V', E')$ constructed for an instance of the 1-distance m-tuple (ℓ, r)-domination problem.

m-tuple (ℓ, r)-dominating set of size at most $k + \ell$.

Proof: Let D be a dominating set of G and $|D| \leq k$. Let $L = \{v_i^1 \mid v_i \in D\} \cup V^2 \cup \{v_1^3\}$. Now, we show that L is a 1-distance m-tuple (ℓ, r)-dominating set in G'.

(i) Observe that for each $v \in V'$, $|N_{G'}[v] \cap L| \geq m$ as $m \leq \ell$ (value of $r = 1$ in case of $m = \ell$) and each $v \in V'$ is dominated by $\ell - 1$ vertices in V^2.

(ii) Let $U = \{u_1, u_2, \ldots, u_r\} \subseteq V'$ be an arbitrary subset of size r.

Case 1: Let $U \cap V^2 \neq \emptyset$ and $v_i^2 \in U \cap V^2$. From the construction of G', $N_G[v_i^2] \cap L \supseteq V^2 \cup \{v_1^3\}$, which implies $|N_G[v_i^2] \cap L| \geq \ell$. Therefore, $|(\cup_{u \in U} N_G[u]) \cap L| \geq \ell$.

Case 2: Let $U \cap V^1 \neq \emptyset$ and $v_i^1 \in U \cap V^1$. From the constructions of G' and L, $N_G[v_i^1] \cap L \supseteq V^2 \cup \{v_j^1\}$, where $v_j \in D$ is a dominator of v_i in G. Therefore, $|N_G[v_i^1] \cap L| \geq \ell$, which leads to $|(\cup_{u \in U} N_G[u]) \cap L| \geq \ell$.

Case 3: Let $U = V^3$. Again, from the constructions of G and L, $(\cup_{u \in U} N_G[u]) \cap L \supseteq V^2 \cup \{v_1^3\}$. Therefore, in this case also $|(\cup_{u \in U} N_G[u]) \cap L| \geq \ell$.

Thus, L is a 1-distance m-tuple (ℓ, r)-dominating set in G' and $|L| \leq k + \ell$.

Conversely, let L be a 1-distance m-tuple (ℓ, r)-dominating set for G' of size at most $k+\ell$. From the definition of the 1-distance m-tuple (ℓ, r)-dominating set (see Definition 4.0.1) and as $|V^3| = r$, $|(\cup_{v \in V^3} N_{G'}[v]) \cap L| \geq \ell$. Therefore, there must be at least ℓ vertices from $V^2 \cup V^3$ in L (see Figure 4.1). Let $D = \{v_i \in V \mid v_i^1 \in L \setminus (V^2 \cup V^3)\}$. If D is a dominating set of G, then we are done as $|D| \leq k$. Suppose D is not a dominating set in G. Since $|V^2|$ is $\ell - 1$, the 1-distance neighborhood of every subset of V^1 with cardinality greater than or equal to r will have a non-empty intersection with D (due to the second condition of 1-distance m-tuple (ℓ, r)-domination). This implies, for any subset U^1 of V, $D \cap (\cup_{u \in U^1} N_G[u]) = \emptyset$ if and only if $|U^1| \leq r - 1$. Note that, such a set $U^1 (\neq \emptyset)$ exists based on our assumption that D is not a dominating set of G. Let $|U^1| = s$. Now, we will show that $|(V^2 \cup V^3) \cap L| \geq \ell + s$.

Let $U^2 (\subseteq V^2)$ and $U^3 (\subseteq V^3)$ be the maximum size subsets such that $U^2 \cap L = \emptyset$ and $U^3 \cap L = \emptyset$, respectively. Let $s' = |U^2|$ and $s'' = |U^3|$. Let $U_{13} = U^1 \cup U^3$. Since $(\cup_{u \in U_{13}} N_{G'}[u]) \cap L = V^2 \setminus U^2$, i.e., $|(\cup_{u \in U_{13}} N_{G'}[u]) \cap L| = \ell - 1 - s' < \ell$, $|U_{13}| = s + s'' < r$. Add $r - s - s''$ vertices from $V^3 \setminus U^3$ to the vertex set U_{13}. Now, by the definition of L the size of the set $|(\cup_{u \in U_{13}} N_{G'}[u]) \cap L|$ must be at least ℓ. Therefore, $r - s - s'' + \ell - 1 - s' \geq \ell$, which implies $r - s'' \geq s + s' + 1$.

Since $r - s'' \geq s + s' + 1$, $|D| \leq k - s$. Let $D_1 = D \cup U^1$. So, every vertex in V is dominated by at least one vertex in D_1 whose size is at most k. Therefore, we conclude, the decision version of 1-distance m-tuple (ℓ, r)-domination problem belongs to the class NP-complete. □

4.1.2 Hardness Result of the d-Distance m-Tuple $(\ell, 2)$-Domination Problem

In this section, we prove that the decision version of d-distance m-tuple $(\ell, 2)$-domination problem belongs to the class NP-complete. For fixed constant $d \geq 2$, the decision version of the problem is defined as follows.

Definition 4.1.4. *Given a simple undirected graph $G = (V, E)$ with $|V| \geq \ell$ and three positive integers m, d, and $k(\leq |V|)$, where $m \leq \ell$, does there exist a d-distance m-tuple $(\ell, 2)$-dominating set of size at most k?*

We prove that, the decision version of the d-distance m-tuple $(\ell, 2)$-domination problem ($d \geq 2$) belongs to the class NP-complete by reducing the decision version of the 1-distance m-tuple $(\ell, 2)$-domination problem to it in polynomial time. Note that 1-distance m-tuple $(\ell, 2)$-domination problem belongs to the class NP-complete (see Section 4.1.1). Recall, the decision version of 1-distance m-tuple $(\ell, 2)$-domination problem:

> *Given a simple undirected graph $G = (V, E)$ with $|V| \geq \ell$ and two positive integer $m, k \leq |V|$, where $m \leq \ell$, does there exist a 1-distance m-tuple $(\ell, 2)$-dominating set of size at most k?*

Theorem 4.1.5. *The decision version of the d-distance m-tuple $(\ell, 2)$-domination problem belongs to the class NP-complete.*

Proof. The decision version of the d-distance m-tuple $(\ell, 2)$-domination problem belongs to the class NP as for a given certificate (a subset of V) we can verify whether it is satisfying both the conditions of the d-distance m-tuple $(\ell, 2)$-domination or not in polynomial-time.

We now describe a polynomial-time reduction from an arbitrary instance of the decision version of 1-distance m-tuple $(\ell, 2)$-domination problem to an instance of the decision version of the d-distance m-tuple $(\ell, 2)$-domination problem.

Let $G = (V = \{v_1, v_2, \ldots, v_n\}, E)$ be an arbitrary instance of the decision version of 1-distance m-tuple $(\ell, 2)$-domination problem. We construct an instance, a graph $G' = (V', E')$, of the decision version of the d-distance m-tuple $(\ell, 2)$-domination problem as follows:

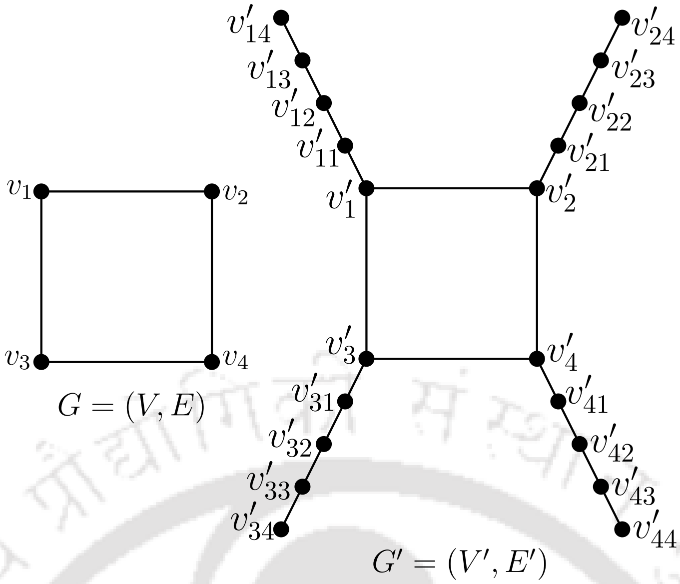


Figure 4.2: A graph $G' = (V', E')$ constructed for an instance of the d-distance m-tuple $(\ell, 2)$- domination problem.

$$V' = \{v'_i \mid v_i \in V\} \cup (\bigcup_{v_i \in V} \{v'_{i1}, v'_{i2}, \ldots, v'_{id-1}\})(see\ Figure\ 4.2\ for\ an\ example)$$

$$E' = \{v'_i v'_j \mid v_i v_j \in E\} \cup (\bigcup_{v_i \in V} \{v'_i v'_{i1}, v'_{i1} v'_{i2}, \ldots, v'_{id-2} v'_{id-1}\})$$

Claim 2: *G has a 1-distance m-tuple $(\ell, 2)$-dominating set of cardinality at most k if and only if G' has a d-distance m-tuple $(\ell, 2)$-dominating set of cardinality at most k.*

Necessity: Let L be a 1-distance m-tuple $(\ell, 2)$-dominating set of G such that $|L| \leq k$. Let $L' = \{v'_i \in V' \mid v_i \in L\}$. We can argue that L' is a d-distance m-tuple $(\ell, 2)$-dominating set in G' and $|L'| \leq k$. Since $|L'| = |L|$ and $|L| \leq k$, so $|L'| \leq k$. As each vertex $v \in V$ satisfies 1-distance m-tuple $(\ell, 2)$-domination properties and each vertex in G' is at most $d - 1$ distance away from a vertex in L', L' suffices to ensure d-distance m-tuple $(\ell, 2)$-domination in graph G' for $d \geq 2$.

Sufficiency: Let L' be a d-distance m-tuple $(\ell, 2)$-dominating set in G' such that $|L'| \leq k$. We shall show that, by updating (i.e., removing or replacing) some of the vertices in L', at most k vertices from $\{v'_1, v'_2, \ldots, v'_n\}$ can be chosen such that the set of corresponding vertices in V is an 1-distance m-tuple $(\ell, 2)$-dominating set in G.

Let $L'' = L'$. For each vertex $v'_{ij} \in V'$, $(1 \leq j \leq d-1$ and $1 \leq i \leq n)$ we do the following:

if $v'_{ij} \in L''$, then replace it with its associated vertex v'_i if v'_i is not already in L'', otherwise, replace it with any vertex in $N_{G'}[v'_i] \cap \{v'_1, v'_2, \ldots, v'_n\}$ which is not in L''. If all the vertices of $N_{G'}[v'_i] \cap \{v'_1, v'_2, \ldots, v'_n\}$ are in L'' (i.e., $(N_{G'}[v'_i] \cap \{v'_1, v'_2, \ldots, v'_n\}) \subseteq L''$), then remove v'_{ij} from L''. Therefore, $|L''| \leq k$. Let $L = \{v_i \in V \mid v'_i \in L''\}$. Now, we prove that L is an 1-distance m-tuple $(\ell, 2)$-dominating set in G such that $|L| \leq k$.

Since $|L''| \leq k$, then $|L| \leq k$. We first prove the first condition (i.e., for every $v \in V$, $|N_G[v] \cap L| \geq m$) of 1-distance m-tuple $(\ell, 2)$-dominating set. Consider a vertex $v'_i \in V'$, for some $1 \leq i \leq n$, let s be the number of vertices in $L' \cap \{v'_{i1}, v'_{i2} \ldots, v'_{id-1}\}$.

Case 1. $s = 0$. Since L' is d-distance m-tuple $(\ell, 2)$-dominating set, there must exist at least m vertices, say $\{v''_1, v''_2, \ldots, v''_m\}$ in $\{v'_1, v'_2, \ldots, v'_n\} \cap L'$ such that $\{v''_1, v''_2, \ldots, v''_m\} \subseteq N^d_{G'}[v'_{i,d-1}]$, otherwise, L' is not a feasible solution as v'_{id-1} does not have m distance-d dominators. Therefore, $|N_{G'}[v'_i] \cap (\{v'_1, v'_2, \ldots, v'_n\} \cap L'')| \geq m$.

Case 2. $s \geq 1$. Let $v'_{ij_1}, v'_{ij_2}, \ldots, v'_{ij_t} \in L'$, for some $1 \leq j_1, j_2, \ldots, j_t \leq d-1$. By our construction of L'' each vertex in $\{v'_{ij_1}, v'_{ij_2}, \ldots, v'_{ij_t}\}$ is replaced by one of the vertices in $N_{G'}[v'_i] \cap \{v'_1, v'_2, \ldots, v'_n\}$. Therefore, in this case also $|N_{G'}[v'_i] \cap (\{v'_1, v'_2, \ldots, v'_n\} \cap L'')| \geq m$. Thus, by our construction of L from L'', $|N_G[v_i] \cap L| \geq m$ is true.

Now we prove the second condition of 1-distance m-tuple $(\ell, 2)$-dominating set (i.e., for every pair of distinct vertices $u, v \in V$, $|(N_G[u] \cup N_G[v]) \cap L| \geq \ell$).

Let v_i and v_j be two distinct vertices in G. Consider the vertices v'_{id-1} and v'_{jd-1} in G'. As L' is a d-distance m-tuple $(\ell, 2)$-dominating set of G', it satisfies the second property of d-distance m-tuple $(\ell, 2)$-domination in G'. Thus there exist at least ℓ dominators dominating v'_{id-1} and v'_{jd-1} in L', i.e., $|(N^d_{G'}[v'_{id-1}] \cup N^d_{G'}[v'_{jd-1}]) \cap L'| \geq \ell$. These dominators are either from $N_G[v'_i] \cup N_G[v'_j]$ or from $\{v'_{i1}, v'_{i2}, \ldots, v'_{id-1}\}$ and/or from $\{v'_{j1}, v'_{j2} \ldots, v'_{jd-1}\}$. As per our construction of L'' from L', we are replacing each dominator in $\{v'_{i1}, v'_{i2}, \ldots, v'_{id-1}\} \cup \{v'_{j1}, v'_{j2} \ldots, v'_{jd-1}\}$ (if any) by a vertex in $(N_{G'}[v'_i] \cup$

$N_{G'}[v'_j]) \cap \{v'_1, v'_2, \ldots, v'_n\}$.

Since G is connected and $|V| \geq \ell$, so is G'. Therefore, L'' contains at least ℓ vertices from $(N_{G'}[v'_i] \cup N_{G'}[v'_j]) \cap \{v'_1, v'_2, \ldots, v'_n\}$, i.e., $|(N_{G'}[v'_i] \cup N_{G'}[v'_j]) \cap \{v'_1, v'_2, \ldots, v'_n\} \cap L''| \geq \ell$. Therefore, according to the construction of L from L'', $|(N_G[v_i] \cup N_G[v_j]) \cap L| \geq \ell$. Thus, L is a 1-distance m-tuple $(\ell, 2)$-dominating set of the graph G having cardinality at most k.

Therefore, the decision version of d-distance m-tuple $(\ell, 2)$-domination problem belongs to the class NP-complete. □

4.2 Inapproximability Results

4.2.1 Inapproximability Result of the 1-Distance m-Tuple (ℓ, r)-Domination Problem

In this section, we prove that the 1-distance m-tuple (ℓ, r)-domination problem cannot be approximated within a factor of $(\frac{1}{2} - \varepsilon)\ln(|V|)$ for any $\varepsilon > 0$, unless P = NP. We argue the claim by showing that if 1-distance m-tuple (ℓ, r)-domination problem can be approximated within a factor of $(\frac{1}{2} - \varepsilon)\ln(|V|)$ for any $\varepsilon > 0$ in a graph G', then the domination problem can be approximated within a factor of $(1 - \varepsilon)\ln(|V|)$ for any $\varepsilon > 0$.

Theorem 4.2.1. *[35] For every $\varepsilon > 0$, it is NP-hard to approximate set cover problem within a factor of $(1 - \varepsilon)\ln n$, where n is the size of the instance. The reduction runs in $n^{O(1/\varepsilon)}$ time.*

Theorem 4.2.2. *Minimum domination problem cannot be approximated within a factor of $(1 - \varepsilon)\ln(|V|)$ for any $\varepsilon > 0$, unless P = NP.*

Proof. The result follows from (i) the relation between set cover problem and dominating set problem, (ii) Theorem 4.2.1, and (iii) the inapproximability result in [23]. □

Theorem 4.2.3. *Minimum 1-distance m-tuple (ℓ, r)-domination problem cannot be approximated within a factor of $(\frac{1}{2} - \varepsilon)\ln(|V|)$ for any $\varepsilon > 0$, unless P = NP.*

Proof. Let G be a simple undirected graph. Consider the construction of the graph G' for any given graph G as discussed in Section 4.1.1. As per our construction, we prove that each instance of domination problem can be reducible to an instance of 1-distance m-tuple (ℓ, r)-domination problem in polynomial-time .

Let D^* and L^* be the optimal DS and 1-distance m-tuple (ℓ, r)-dominating set in G and G', with cardinalities $\gamma_{ds}(G)$ and $\gamma_{m\ell r}(G')$, respectively. Now we can argue the following claim: $\gamma_{m\ell r}(G') = \gamma_{ds}(G) + \ell$. The inequality $\gamma_{m\ell r}(G') \leq \gamma_{ds}(G) + \ell$ is trivial as per our construction in Section 4.1.1. On the other hand, $\gamma_{m\ell r}(G') \geq \gamma_{ds}(G) + \ell$ follows from the sufficiency proof of Claim 1 in Section 4.1.1. So given a dominating set D of G, one can find a 1-distance m-tuple (ℓ, r)-dominating set L of G' such that $|L| = |D| + \ell$. Now, $\frac{|L|}{|L^*|} = \frac{|D|+\ell}{|D^*|+\ell} \geq \frac{1}{2}\frac{|D|}{|D^*|}$. Suppose there exist a polynomial time algorithm that approximates 1-distance m-tuple (ℓ, r)-domination problem within a factor of $(\frac{1}{2}-\varepsilon)\ln N$ for graphs with N vertices. As per our construction of the graph G' from G (see Figure 4.1), G' contains, $N = n + \ell + r - 1 \leq 3n$ for $n \geq 2$ vertices, where n is the total number of vertices in G, $\ell < n$, and $r < n$. Therefore,

$$\frac{|D|}{|D^*|} \leq (1 - 2\varepsilon)\ln N \leq (1 - 2\varepsilon)\ln n(1 + \frac{\ln 4}{\ln n}).$$

For sufficiently large n, the term $(1 + \frac{\ln 4}{\ln n})$ can be bounded by $1 + \frac{\varepsilon}{5}$, where $\varepsilon \geq \frac{5\ln 4}{\ln n}$. Now we have

$$(1 - 2\varepsilon)\ln n(1 + \frac{\ln 4}{\ln n}) \leq (1 - 2\varepsilon)[\ln n + \ln(1 + \frac{\varepsilon}{5})] \leq (1 - 2\varepsilon)[\ln n + \frac{\varepsilon}{5}\ln n] \leq (1 - \varepsilon')\ln n,$$

where $\varepsilon' < \frac{9}{5}\varepsilon + \frac{2}{5}\varepsilon^2$. Therefore, for an arbitrary graph, we can approximate the domination problem by a factor of $(1 - \varepsilon')\ln n$, which leads to a contradiction to Theorem 4.2.2. Thus, the minimum 1-distance m-tuple (ℓ, r)-domination problem cannot be approximated within a factor of $(\frac{1}{2} - \varepsilon)\ln(|V|)$ for any $\varepsilon > 0$, unless P = NP. □

4.2.2 Inapproximability Result of the d-Distance m-Tuple $(\ell, 2)$-Domination Problem

In this section, we give a lower bound on the approximation ratio of any approximation algorithm for the d-distance m-tuple $(\ell, 2)$-domination problem by providing an approximation preserving reduction from the 1-distance m-tuple (ℓ, r)-domination problem for $r = 2$.

Theorem 4.2.4. *Given a simple undirected graph $G = (V, E)$, the d-distance m-tuple $(\ell, 2)$-domination problem cannot be approximated within a factor of $(\frac{1}{4} - \varepsilon) \ln |V|$, for any fixed constant $d \geq 2$ and $\varepsilon > 0$, unless P = NP.*

Proof. Let $G = (V, E)$ be an arbitrary instance of the 1-distance m-tuple $(\ell, 2)$-domination problem with n vertices. Given $G = (V, E)$, we construct a graph $G' = (V', E')$, an instance of the d-distance m-tuple $(\ell, 2)$-domination problem as described in Section 4.1.2. Let L^* and L_d^* be the optimal 1-distance m-tuple $(\ell, 2)$-dominating set and d-distance m-tuple $(\ell, 2)$-dominating set in G and G', with cardinalities $\gamma_{m\ell}(G)$ and $\gamma_{m\ell}^d(G')$, respectively. Now we can argue the following claim: $\gamma_{m\ell}^d(G') = \gamma_{m\ell}(G)$. The inequality $\gamma_{m\ell}^d(G') \leq \gamma_{m\ell}(G)$ is trivial as every 1-distance m-tuple $(\ell, 2)$-dominating set of G is a d-distance m-tuple $(\ell, 2)$-dominating set in G'. On the other hand, $\gamma_{m\ell}^d(G') = |L_d^*| \geq |L|$ follows from the sufficiency proof of Claim 2 in Section 4.1.2.

Given any 1-distance m-tuple $(\ell, 2)$-dominating set L of G, one can find a d-distance m-tuple $(\ell, 2)$-dominating set L_d of G' with $|L_d| = |L|$. Suppose there exist a polynomial time algorithm to approximate d-distance m-tuple $(\ell, 2)$-domination problem within a factor of $(\frac{1}{4} - \varepsilon) \ln |V'|$, where $|V'| = n + n(d-1) \leq n^2$ (see Section 4.1.2). Now $\frac{|L|}{|L^*|} = \frac{|L_d|}{|L_d^*|} \leq (\frac{1}{4} - \varepsilon) \ln n^2 = (\frac{1}{2} - 2\varepsilon) \ln n \leq (\frac{1}{2} - \varepsilon') \ln n$, where $\varepsilon' \leq 2\varepsilon$. Therefore, the result follows from Theorem 4.2.3. □

4.3 Conclusion

In this chapter, we studied d-distance m-tuple (ℓ, r)-domination problem. We provided a common NP-completeness proof of the 1-distance m-tuple (ℓ, r) domination problem for each fixed value of m, ℓ, and r. We also presented a common NP-completeness proof of the d-distance m-tuple $(\ell, 2)$ domination problem for each fixed value of $d(> 1), m$, and ℓ. We have showed that the first problem is not approximated within a factor of $(\frac{1}{2} - \varepsilon) \ln |V|$ for each fixed value of m, ℓ, and r, unless P = NP and the second problem is not approximated within a factor of $(\frac{1}{4} - \varepsilon) \ln |V|$ for each fixed value of $d(> 1), m$, and ℓ, unless P = NP, where V is the vertex set of the input graph. The reduction in the NP-completeness/inapproximability proofs are very powerful as these are common reductions for completely different variant of dominations.

Chapter 5

Vertex-Edge Dominating Set Problem in Unit Disk Graphs

The vertex-edge dominating set (VEDS) problem is one of the variants of the dominating set problem. The VEDS problem is widely studied in different graph classes in the literature. To the best of our knowledge, we are the first to define the problem in unit disk graphs. In this chapter, we study the geometric minimum vertex-edge dominating set (GMVEDS) problem, defined as follows:

Definition 5.0.1. (Problem definition) *Given a simple unit disk graph $G = (V, E)$ corresponding to a point set $P = \{p_1, p_2, \ldots, p_n\}$ for disk centers in the plane, find a minimum cardinality subset $D \subseteq V$ such that every edge $e \in E$ is vertex-edge dominated by at least one vertex in D.*

An edge $e = uv \in E$ is said to be vertex-edge dominated by a vertex $w \in V$ if w is in closed neighborhood of either u or v i.e., $w \in N_G[u] \cup N_G[v]$, where the closed neighborhood of a vertex v in a simple undirected graph $G = (V, E)$ is defined as $N_G[v] = \{u \in V \mid uv \in E\} \cup \{v\}$.

In a vertex prospective, we can say a vertex $v \in V$, vertex-edge dominates every edge uv, as well as every edge adjacent to these edges.

We first prove that the decision version of the GMVEDS problem belongs to the NP-complete class in unit disk graphs. Next, we present a simple 4-factor approximation algorithm for the GMVEDS problem. Finally, we propose a PTAS for the same problem.

5.1 NP-hardness Result of the GMVEDS Problem

In this section, we show a polynomial-time reduction from the NP-hard *vertex cover* problem in planar graphs [43] to the GMVEDS problem to prove that the latter one is also NP-hard. The decision versions of both these problems are defined below.

Definition 5.1.1. (The VEDS problem in UDGs (Veds-Udg)) *Given a unit disk graph $G = (V, E)$ and a positive integer k, does there exist a vertex-edge dominating set D of G such that $|D| \leq k$?*

Definition 5.1.2. (The vertex cover problem in planar graphs (Vc-Pla)) *Given a planar graph $G = (V, E)$ having maximum degree 3 and a positive integer k, does there exist a vertex cover C of G such that $|C| \leq k$*

Corollary 5.1.3. *A planar graph $G = (V, E)$ with maximum degree 3 and $|V| \geq 3$ can be embedded in the plane with its vertices are at $(4i, 4j)$ and its edges are drawn as a sequence of consecutive line segments on the lines $x = 4i$ or $y = 4j$, for some i and j* (see Figure 5.1).

Proof. Follows from Lemma 3.1.4. □

This embedding is known as the *orthogonal drawing* of a graph. Using Lemma 3.1.6, an orthogonal drawing of the given graph can be produced with at most 2 bends along each edge in linear time (see Figure 5.1).

Lemma 5.1.4. *Let $G = (V, E)$ be an instance of* Vc-Pla *with $|E| \geq 2$. An instance $G' = (V', E')$ of* Veds-Udg *can be constructed from G in polynomial-time.*

Proof. Construction of G' from G is done in three steps.

Step 1: (Embedding graph G into a grid of size $4n \times 4n$) G can be embedded in

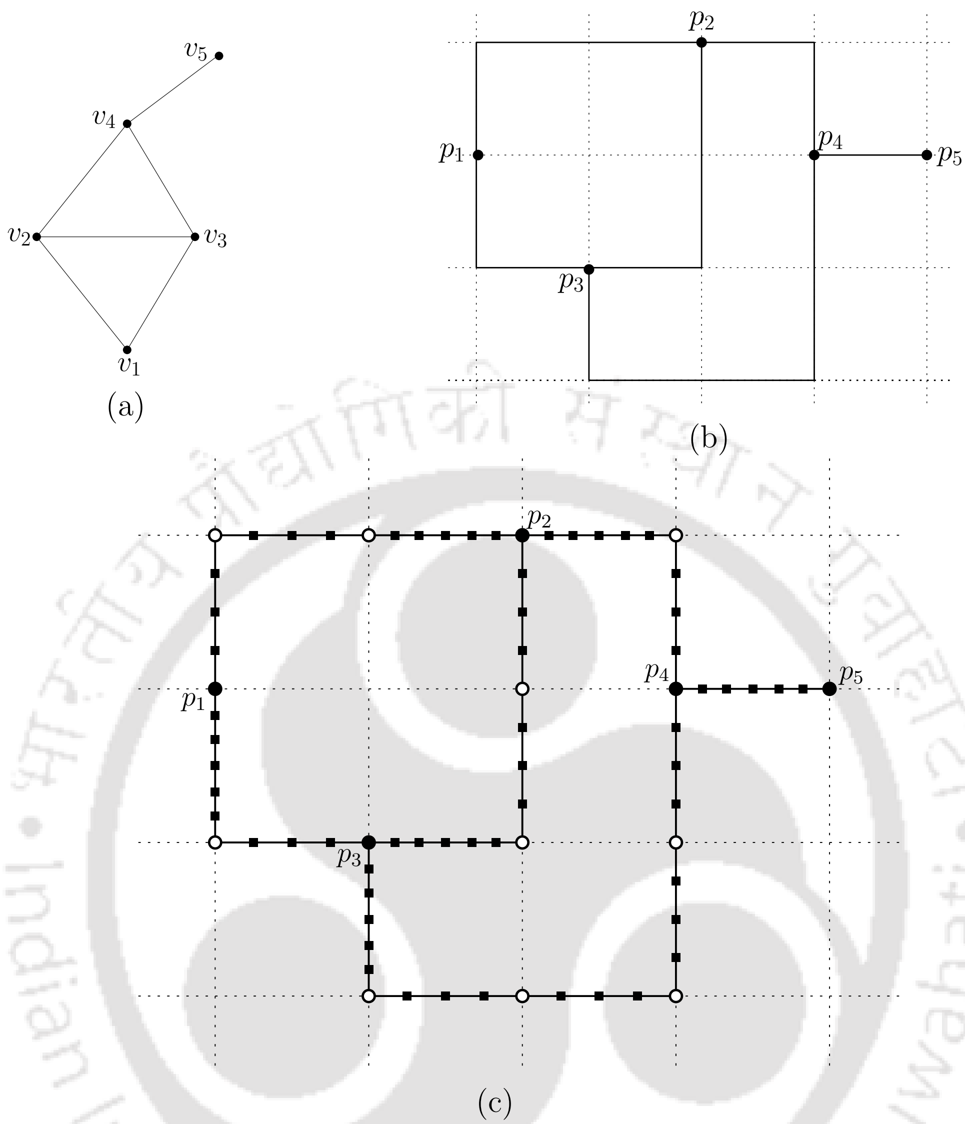


Figure 5.1: (a) A planar graph G, (b) embedding of G on a grid, and (c) a UDG construction from the embedding.

the plane using one of the algorithms [56, 59] (see Corollary 5.1.3) with each of its edges as a sequence of a connected line segment(s) of length four units. Let the total number of line segments used in the embedding is ℓ. The points $\{p_1, p_2, \ldots, p_n\}$ are termed *node points* in the embedding correspond to the vertex set $V = \{v_1, v_2, \ldots, v_n\}$ (see Figure 5.1(a) and 5.1(b)).

Step 2: (Adding extra points) For each edge $p_i p_j$ having length 4 units, (i) we add two points α and β on the edge $p_i p_j$ such that α is 0.8 unit appart from p_i and β is 0.8

unit appart from p_j, and (ii) add another three points between α and β with distance 0.6 unit from each other, respectively (thus adding five points in total, see edge p_4p_5 in Figure 5.1(c)). For each edge of length greater than 4 units, we also add points as follows: (i) add a point in the joining point (grid point) of each line segments other than the *node points* and name it as a *joint point* (see empty circular points in Figure 5.1(c)), (ii) for one of the two-line segments whose one endpoint is associated with node point, we add five points at distances 0.8, 1.4, 2, 2.6, and 3.2 units from the node point, and for other line segments, we add three points at distance 1 units from each other excluding the *joint points* (see the edge p_3p_4 in Figure 5.1(c)). We name the points added in this step as *added points*.

Step 3: (Construction step) For convenience, denote the set of node points by N and set of added points by A, respectively, that is, $N = \{p_i \mid v_i \in V\}$ and $A = \{q_1, q_2, \ldots, q_{4\ell+|E|}\}$. We construct a UDG $G' = (V', E')$, where $V' = N \cup A$ and there is an edge between two points in V' if and only if the Euclidean distance between the points is at most 1 (see Figure 5.1(c)). Observe that $|N| = |V|(= n)$ and $|A| = 4\ell + |E|$, where ℓ is the total number of line segments in the embedding and $|E|$ is the total number of edges in G. Since G is planar, $|E| = O(n)$. It also follows from Corollary 5.1.3 that $\ell = O(n^2)$. Therefore both $|V'|$ and $|E'|$ are bounded by $O(n^2)$, and hence G' can be constructed in polynomial-time. □

Theorem 5.1.5. VEDS-UDG *belongs to the class NP-complete.*

Proof. For any given set $D \subseteq V$ and a positive integer k, we can verify in polynomial-time whether D is a vertex-edge dominating set of size at most k by checking whether each edge in E is vertex-edge dominated by a vertex in D or not. Hence, VEDS-UDG $\in$ NP.

Now, we need to prove VEDS-UDG $\in$ NP-hard. For the hardness proof, we show a polynomial-time reduction from VC-PLA to VEDS-UDG. Let $G = (V, E)$ be an instance of VC-PLA. Construct the instance $G' = (V', E')$ of VEDS-UDG as discussed in Lemma 5.1.4. We have the following claim.

Claim. *G has a vertex cover of size at most k if and only if G' has a vertex-edge dominating set of size at most $k + \ell$.*

Necessity. Let $C \subseteq V$ be a vertex cover of G such that $|C| \leq k$. Let $N' = \{p_i \in N \mid v_i \in C\}$, i.e., N' is the set of vertices in G' that correspond to the vertices in C. The idea is to choose one vertex from each segment in the embedding such that the chosen vertex set $A'(\subseteq A)$ together with N', i.e., $N' \cup A'$ will form a VEDS of cardinality $k + \ell$ in G'. As C is a vertex cover in G, every edge in G has at least one of its endpoints in C. Let v_iv_j be an edge in G and assume $v_i \in C$ (the same argument works for $v_j \in C$ or if both v_i and $v_j \in C$). It follows from the construction of G' that the edge p_ip_j is represented as a sequence of line segments in the graph G', where p_i and p_j are nodes in G' corresponding to vertices v_i and v_j in G. Start traversing the segments from p_i, and add each fourth vertex to A' encountered from p_i to p_j in the traversal (see Figure 5.2 for an illustration, where both big circles and squares belong to A' while traversing from p_1 to p_2, p_2 to p_3, p_1 to p_3, p_4 to p_2, p_4 to p_5 and p_4 to p_3, respectively).

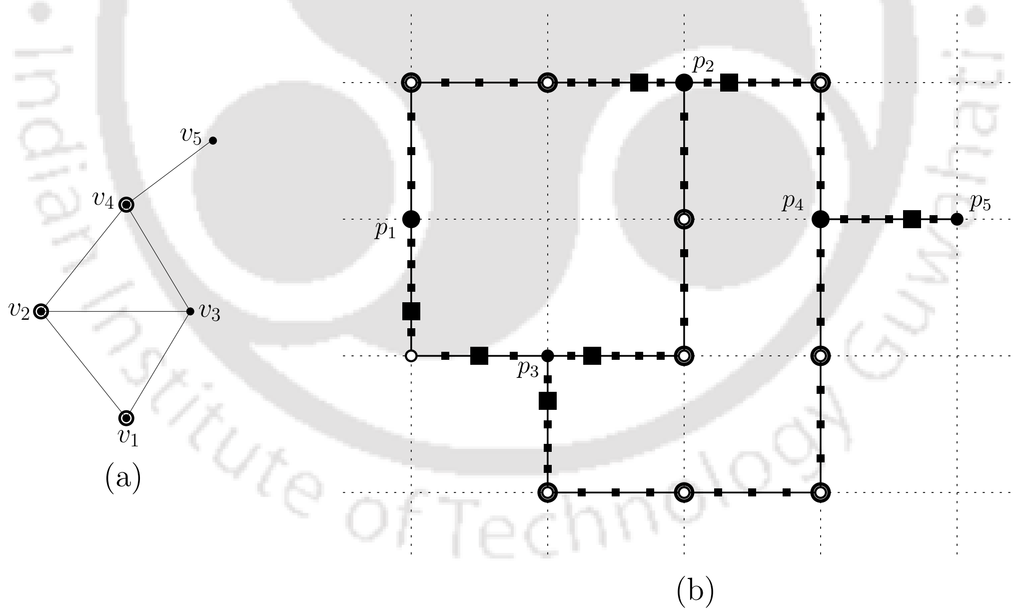


Figure 5.2: (a) A vertex cover $\{v_1, v_2, v_4\}$ of G, and (b) the construction of A' in G'

Apply the same process to each chain of line segments in G' corresponding to each edge in G. The cardinality of A' is ℓ as we have chosen one vertex from each segment in the embedding. Let $D = N' \cup A'$. Now, observe that D is a vertex-edge dominating set in G' as each edge in G' is vertex-edge dominated by at least one vertex in D and $|D| = |N'| + |A'| \leq k + \ell$ as required.

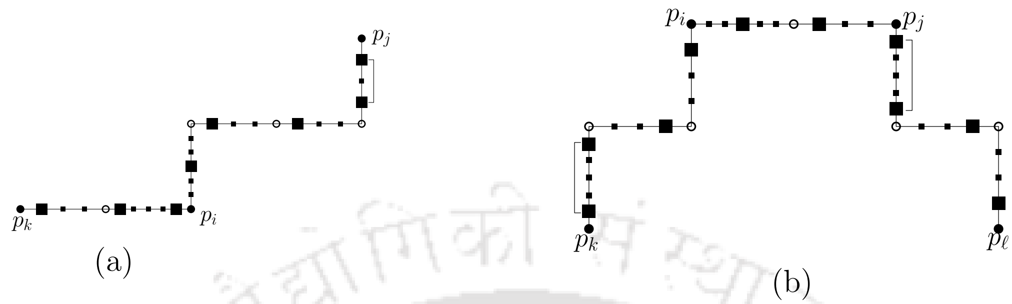


Figure 5.3: (a) p_i is connected with only p_j, and (b) p_i is connected with p_k and p_j is connected with p_ℓ.

Sufficiency. Let $D \subseteq V'$ be a VEDs of size at most $k + \ell$. We argue that G has a vertex cover of size at most k based upon the following **claim:**

- at least one vertex on each segment in the embedding must belong to D and hence $|A \cap D| \geq \ell$, where ℓ is the total number of segments in the embedding.

We shall show that, by removing and/or replacing some vertices in D, a set of size at most k vertices from N can be chosen such that the corresponding vertices in G is a vertex cover. Let $C = \{v_i \in V \mid p_i \in D \cap N\}$. If any edge v_iv_j in G has none of its end vertices in C, then consider the points p_i and p_j corresponding to v_i and v_j respectively.

Case (i): If p_i is the only vertex that is connected with p_j in G', then the chain of segments (say ℓ') in the path $p_i \rightsquigarrow p_j$ in G' has at least $\ell' + 1$ vertices in D (see Figure 5.3(a) for example). In this case, we delete one point from the segment containing two points in D and introduce p_i in D.

Case (ii): If both p_i and p_j are connected with some points p_k and p_ℓ respectively in G', then either

- the chain of segments (say ℓ') in the path $p_i \rightsquigarrow p_j$ in G' has at least $\ell' + 1$ vertices in D (similar to Case (i)) or

- the chain of segments (say ℓ') in both the path $p_i \rightsquigarrow p_k$ and $p_j \rightsquigarrow p_\ell$ in G' has at least $\ell' + 1$ vertices in D (see Figure 5.3(b) for example).

In this case,

Case (a) : If p_i is the only vertex that is connected with p_k, such that $p_k \notin D$ (similar case applies if p_j is the only vertex connected with p_ℓ and $p_\ell \notin D$) , then we choose the segment from $p_i \rightsquigarrow p_k$ having two vertices in D and remove one vertex of the segment from D and introduce p_i in D.

Case (b) : If $p_k \in D$ (similar case applies if $p_\ell \in D$), then remove one vertex from the segment having its two vertices in D in the path $p_i \rightsquigarrow p_k$ and introduce p_j in D.

Case (c) : If Case (a) and (b) fails, then arbitrarily choose p_i or p_j in D by removing one vertex from the segment containing its two vertices in D from the path $p_i \rightsquigarrow p_k$ or $p_j \rightsquigarrow p_\ell$, respectively.

Update C and repeat the process till every edge has at least one of its end vertices in C. Due to Claim (i), C is a vertex cover in G with $|C| \leq k$. Therefore, VEDS-UDG is NP-hard.

As VEDS-UDG is in NP as well as in NP-hard, VEDS-UDG is in NP-complete. □

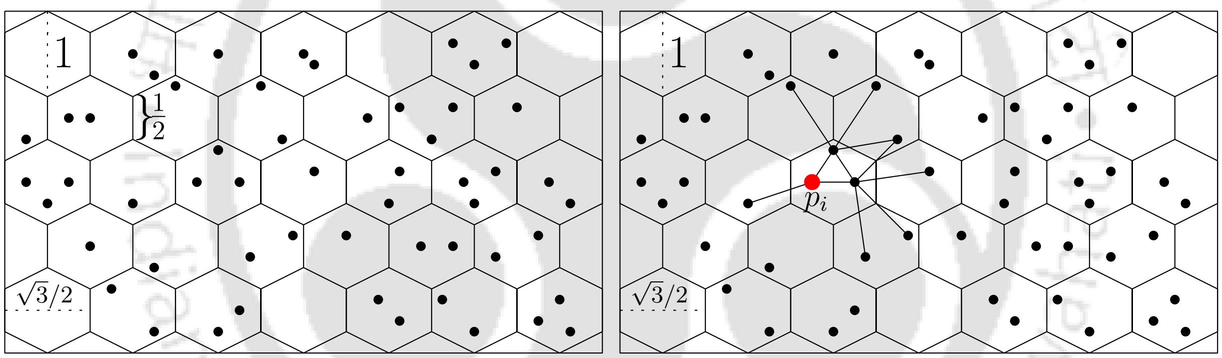
5.2 Approximation Algorithm for GMVEDS Problem

Let $\mathcal{R}$ be an axis parallel rectangular region containing the point set $P = \{p_1, p_2, \cdots, p_n\}$ (center of the UDGs) and the edge set $E = \{e_1, e_2, \cdots, e_m\}$. We add an edge between two points if and only if the distance between two points is at most 1. We present a 4-factor approximation algorithm for the VEDS problem in unit disk graphs with the help of the 4-color partitioning technique given by De et al. [33]. Carmi et al. [15] used a similar type of technique to obtain a 4-factor approximation algorithm for minimum dominating set problem in unit disk graphs, where they used the hexagonal partitioning of $\mathcal{R}$ instead of the square partitioning given by De et al. [33] to improve the time complexity.

Here, we partition $\mathcal{R}$ into regular hexagonal *cells*, where the side length of each regular hexagonal cell is $\frac{1}{2}$ (see Figure 5.4 (a)). After partitioning $\mathcal{R}$, we use the 4-color partitioning technique on it to propose a 4-factor approximation algorithm for the VEDS problem in unit disk graphs. The proposed algorithm runs in polynomial-time.

Claim (i): As the distance between any pair of points inside a cell is at most 1, any point chosen from a cell can vertex-edge dominate all the edges of that cell (see Figure 5.4 (b), p_i vertex-edge dominates all the edges of the cell containing p_i).

By referring to all the edges of a cell, we mean, each edge which both of the endpoints are within the cell or one of the endpoints is within the cell (see all the edges shown in Figure 5.4 (b), these edges are within the cell containing p_i).

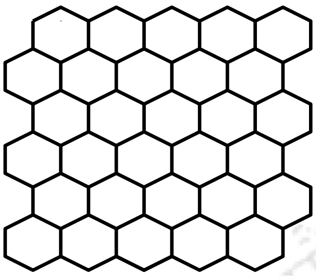


(a) (b)

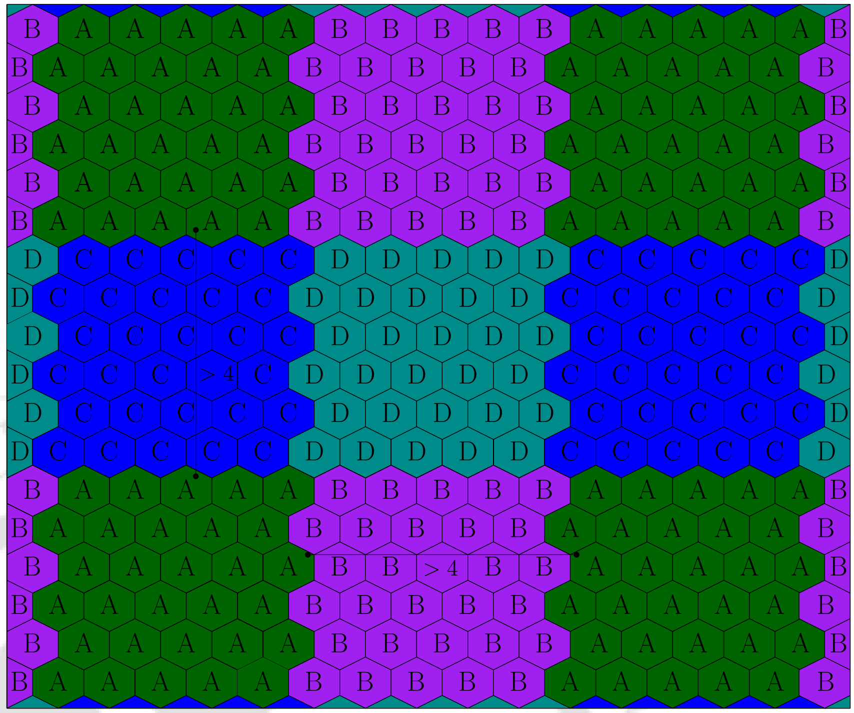
Figure 5.4: (a) $\mathcal{R}$ containing the point set P partitioned into regular hexagons, and (b) a point p_i vertex-edge dominate the edges in the cell containing p_i.

We define a *combined-cell* as a combination of 30 regular hexagonal cells such that it is a combination of 6 consecutive rows and each row consist of 5 consecutive hexagonal cells (see Figure 5.5(a)).

The idea behind our algorithm is as follows: consider a combined-cell partitioning of $\mathcal{R}$ such that no point of P lies on the boundary of any combined-cell, and a 4-color partitioning of it (see Figure 5.5 (b)). The 4-color combined-cell partitioning of $\mathcal{R}$ color all combined-cells such that if a combined-cell $\mathcal{S}$ is assigned the color D then its



(a)



(b)

Figure 5.5: (a) A combined-cell, and (b) a combined-cell partition of $\mathcal{R}$ and its 4-coloring scheme.

adjacent combined-cells are assigned different three colors, say A, B, and C, such that pair of opposite combined-cells adjacent to $\mathcal{S}$ are assigned the same color (see Figure 5.5 (b)). Now, consider any two combined-cells $\mathcal{S}'$ and $\mathcal{S}''$ of the same color. As there is no point of P lies in the boundary of any combined-cell, the minimum distance between any two points $p \in \mathcal{S}' \cap P$ and $q \in \mathcal{S}'' \cap P$ is greater than 4 (see Figure 5.5 (b)). This leads to the fact that there does not exist a single point that can simultaneously vertex-edge dominate an edge connecting a point from $\mathcal{S}'$ and another edge connecting a point from $\mathcal{S}''$. We can say from the above fact that if OPT_A denotes a minimum size subset of points in P such that OPT_A is a VEDS of the combined-cells colored with A, then OPT_A is the union of the optimum solutions of the VEDS problems for all the combined-cells colored A. The same result holds for all the other three colors also. Now we consider each combined-cell $\mathcal{S}$ and find the minimum size subset of P, which ensures the VEDS of $\mathcal{S}$. Finally, we report the set T which is the union of the solutions for all combined-cells.

Lemma 5.2.1. *T is a VEDS of $\mathcal{R}$ and $|T| \leq 4|OPT|$, where OPT denotes a minimum size subset of P such that OPT vertex-edge dominates all the edges in $\mathcal{R}$.*

Proof. Observe that $T = OPT_A \cup OPT_B \cup OPT_C \cup OPT_D$ and each OPT_i vertex-edge dominates all the edges in combined-cells colored with i. As all the edges of $\mathcal{R}$ is a part of at least one combined-cell colored with a specific color i and vertex-edge dominated by at least one point of OPT_i, T is a VEDS of $\mathcal{R}$. The cardinality of T follows from the fact that each $|OPT_i| \leq |OPT|$ for all $i = A, B, C, D$. □

Computing an optimum solution for a single combined-cell

Let $\mathcal{S}$ be a combined-cell, $P_1 = P \cap \mathcal{S}$. Let P_2 be a subset of P such that each point of P_2 can vertex-edge dominate at least one edge in $\mathcal{S}$. Surely, $P_1 \subseteq P_2$.

Lemma 5.2.2. *If for any combined-cell $\mathcal{S}$, $OPT_{\mathcal{S}}$ is a subset of P_2 of minimum size such that the points in $OPT_{\mathcal{S}}$ can vertex-edge dominate all the edges in $\mathcal{S}$ and if P_2 consists of no more than n_s points, then $OPT_{\mathcal{S}}$ can be computed in $O(n_s^{30})$ time.*

Proof. From Claim (i) in Section 5.2, we know that selecting any one point from a hexagonal cell ensures vertex-edge domination of all the edges in that cell and the combined-cell $\mathcal{S}$ is the combination of 30 hexagonal cells. So $|OPT_{\mathcal{S}}| \leq 30$.

Now, we consider all possible combinations of points in P_2 of size $i = 1, 2, \ldots, 29$ respectively. For each combination, we check whether all the edges in $\mathcal{S}$ are vertex-edge dominated or not. If for any i, there exist a subset of P_2 of size i that vertex-edge dominates all the edges in $\mathcal{S}$, then that subset is reported and execution stops. If this fails for all $i = 1, 2, \ldots, 29$, then the optimum solution of the combined-cell $\mathcal{S}$ consists of any one point from each cell of $\mathcal{S}$. This algorithm needs at most $O(n_s^{30})$ time. □

Theorem 5.2.3. *The proposed 4-coloring scheme gives a 4-factor approximation algorithm for the GMVEDS problem in polynomial-time.*

Proof. The approximation factor follows from Lemma 5.2.1. The time complexity result of the theorem follows from Lemma 5.2.2. □

5.3 Approximation Scheme for GMVEDS Problem

In this section, we propose a PTAS for the VEDS problem in UDGs. Let $G = (V, E)$ be a given UDG. Our PTAS is based on the concept of m-separated collection of subsets of V for some integer m. Given a graph G, let $d(u, v)$ denote the number of edges on a shortest path between u and v. For V_1, $V_2 \subseteq V$, $d(V_1, V_2)$ is defined as $d(V_1, V_2) = min_{u \in V_1, v \in V_2}\{d(u, v)\}$. We use notations $VED(A)$ and $VED_{opt}(A)$ to denote a vertex-edge dominating set of A ($\subseteq V$) in G and an optimal vertex-edge dominating set of A in G. We also define the closed neighborhood of a set $A \subseteq V$ as $N_G[A] = \bigcup\limits_{v \in A} N_G[v]$ and the r-th neighborhood of a vertex v as $N_G^r[v] = \{u \in V \mid d(u, v) \leq r\}$ in G.

Let $\mathcal{S} = \{\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_k\}$ be a collection of disjoint subsets of vertices in G such that each $\mathcal{S}_i \subset V$ for $i = 1, 2, \ldots, k$. The set $\mathcal{S}$ is reffered as a m-separated collection of vertices if $d(\mathcal{S}_i, \mathcal{S}_j) > m$, for $1 \leq i, j \leq k$ and $i \neq j$ (see Figure 5.6 for a 4-separated collection). Nieberg and Hurink [76] considered 2-separated collection to propose a PTAS for the minimum dominating set problem in unit disk graphs.

Lemma 5.3.1. *If $\mathcal{S} = \{\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_k\}$ is an m-separated collection in a graph $G = (V, E)$, then $\sum\limits_{i=1}^{k} |VED_{opt}(S_i)| \leq |VED_{opt}(V)|$ for each $m \geq 4$.*

Proof. For each $\mathcal{S}_i \in \mathcal{S}$, consider $P_i = \{u \in V \mid v \in S_i \text{ and } d(u, v) \leq 2\}$, for $i = 1, 2, \ldots, k$. Since $m \geq 4$, $P_i \cap P_j = \emptyset$ as $d(S_i, S_j) > m$ for $i \neq j$. Observe that, for each $i = 1, 2, \ldots, k$, $\mathcal{S}_i \subseteq P_i$ and $P_i \cap VED_{opt}(V)$ is a vertex-edge dominating set of $\mathcal{S}_i$. Therefore, $(P_i \cap VED_{opt}(V)) \cap (P_j \cap VED_{opt}(V)) = \emptyset$, and hence, we have $\sum\limits_{i=1}^{k} |(P_i \cap VED_{opt}(V))| \leq |VED_{opt}(V)|$. As $P_i \cap VED_{opt}(V)$ is a vertex-edge dominating set of $\mathcal{S}_i$, for $i = 1, 2, \ldots, k$, and $VED_{opt}(V)$ is a minimum vertex-edge dominating set of the graph G, we obtain $\sum\limits_{i=1}^{k} |VED_{opt}(\mathcal{S}_i)| \leq \sum\limits_{i=1}^{k} |(P_i \cap VED_{opt}(V))| \leq |VED_{opt}(V)|$. □

Lemma 5.3.2. *Let $\mathcal{S} = \{\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_k\}$ be an m-separated collection in a graph $G = (V, E)$, $m \geq 4$, and let $\mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_k$ be subsets of V with $\mathcal{S}_i \subseteq \mathcal{R}_i$ for all $i = 1, 2, \ldots, k$. If there exist $\rho \geq 1$ such that $|VED_{opt}(\mathcal{R}_i)| \leq \rho |VED_{opt}(\mathcal{S}_i)|$ holds for all $i = 1, 2, \ldots, k$,*

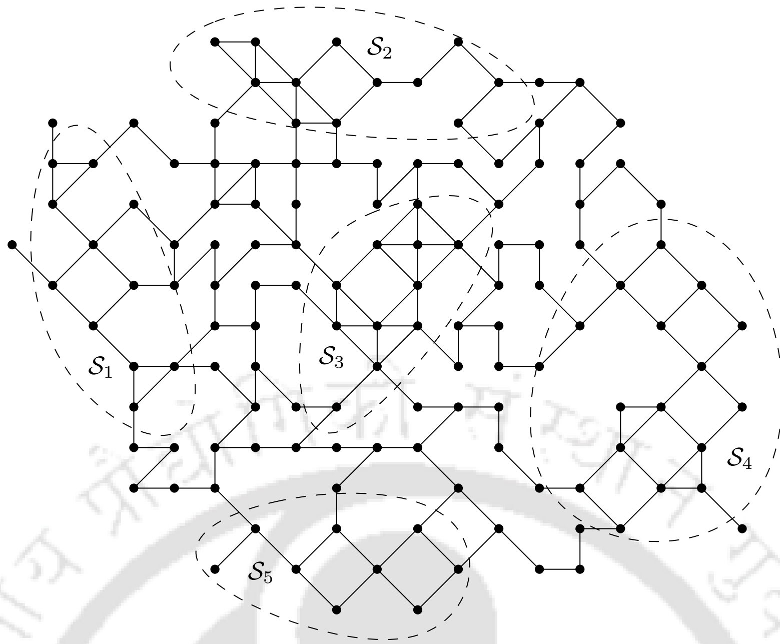


Figure 5.6: A 4-separated collection $\mathcal{S} = \{\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3, \mathcal{S}_4, \mathcal{S}_5\}$

and if $\bigcup_{i=1}^{k} VED_{opt}(\mathcal{R}_i)$ *is a vertex-edge dominating set in* G*, then* $\sum_{i=1}^{k} |VED_{opt}(\mathcal{R}_i)|$ *is at most* ρ *times the size of a minimum vertex-edge dominating set in* G*.*

Proof. $\sum_{i=1}^{k} |VED_{opt}(\mathcal{S}_i)| \leq |VED_{opt}(V)|$ (from Lemma 5.3.1).

Hence, $\sum_{i=1}^{k} |VED_{opt}(\mathcal{R}_i)| \leq \rho \sum_{i=1}^{k} |VED_{opt}(\mathcal{S}_i)| \leq \rho |VED_{opt}(V)|$. □

5.3.1 Construction of subsets

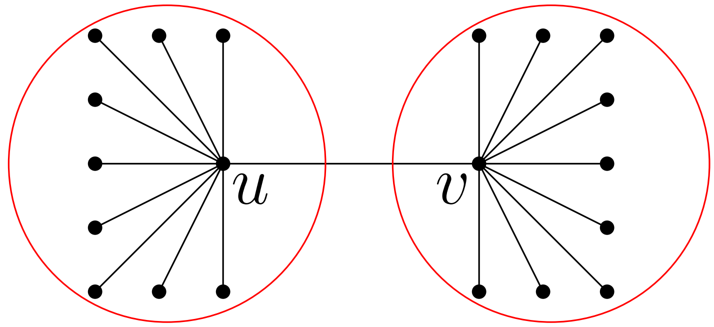
In this section, we discuss the process of constructing the desired 4-separated collection of subsets $\mathcal{S} = \{\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_k\}$ and the corresponding subsets $\mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_k$ of V such that $\mathcal{S}_i \subseteq \mathcal{R}_i$ for all $i = 1, 2, \ldots, k$. The algorithm proceeds in an iterative manner. The basic idea of the algorithm is as follows: start with an arbitrary vertex $v \in V_i$, where V_i is the vertex set in the i-th iteration of the algorithm. Note that, in the first iteration $V_1 = V$ and the algorithm computes $\mathcal{S}_1$ and $\mathcal{R}_1$. More specifically, for $r = 1, 2, \ldots$, we find the vertex-edge dominating set of the graphs induced by the r-th neighborhood as well as the $(r+4)$-th neighborhood of the vertex v until $|VED(N_G^{r+4}[v])| > \rho |VED(N_G^r[v])|$ holds.

Here, $VED(N_G^{r+4}[v])$ and $VED(N_G^r[v])$ are vertex-edge dominating sets of the graph induced by $N_G^{r+4}[v]$ and $N_G^r[v]$, respectively, and $\rho = 1 + \epsilon$ ($\epsilon > 0$). Let $\hat{r}$ be the smallest r violating the above condition. Set $\mathcal{S}_i = N_G^{\hat{r}}[v]$, $\mathcal{R}_i = N_G^{\hat{r}+4}[v]$ and $V_i' = V_i \setminus N_G^{\hat{r}+3}[v]$. Note that removing $N_G^{\hat{r}+4}[v]$ from V_i implies removing the relevant edges connecting $N_G^{\hat{r}+4}[v]$ to $V_i \setminus N_G^{\hat{r}+4}[v]$ for which vertex-edge domination may not be maintained. Hence, removing $N_G^{\hat{r}+3}[v]$ from V_i removes the edges for which $VED(N_G^{\hat{r}+4}[v])$ is a vertex-edge dominating set. Let T_i be the set of vertices consisting of all singleton vertices after removing $N_G^{\hat{r}+3}[v]$ vertices from V_i in the i-th iteration of the algorithm. Set $V_{i+1} = V_i' \setminus T_i$. The process stops while $V_{i+1} = \emptyset$ and returns the sets $\mathcal{S} = \{\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_k\}$ and $\mathcal{R} = \{\mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_k\}$. The collection of the sets $\mathcal{S}$ is a 4-separated collection.

Algorithm 5.1 Vertex-edge_dominating_set (G)

Require: An undirected graph $G = (V, E)$ and an arbitrary small real number $\epsilon > 0$
Ensure: The sets $\mathcal{S} = \{\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_k\}$ and $\mathcal{R} = \{\mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_k\}$
1: $i = 0$, $r = 1$ and $V_{i+1} = V$
2: $\mathcal{S} = \emptyset$, $\mathcal{R} = \emptyset$ and $\rho = 1 + \epsilon$
3: **while** $(V_{i+1} \neq \emptyset)$ **do**
4:     pick an arbitrary $v \in V_{i+1}$
5:     $N^0[v] = v$
6:     $r = 1$
7:     **while** $|(VED(N_G^{r+4}[v])| > \rho |VED(N_G^r[v])|$ **do** ▷ $VED(N_G^r[v]) = MIS(N_G^r[v])$
8:         $r = r + 1$
9:     **end while**
10:     $\hat{r} = r$ ▷ the smallest r violating while condition in step 7
11:     $i = i + 1$ ▷ the index i keeps track of the number of iterations
12:     $S_i = N^{\hat{r}}[v]$
13:     $\mathcal{R}_i = N_G^{\hat{r}+4}[v]$
14:     $V_i' = V_i \setminus N_G^{\hat{r}+3}[v]$
15:     $T_i \leftarrow$ singleton vertices after removing $N_G^{\hat{r}+3}[v]$
16:     $V_{i+1} = V_i' \setminus T_i$
17: **end while**
18: Return $\mathcal{S} = \{\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_k\}$ and $\mathcal{R} = \{\mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_k\}$.

We compute the vertex-edge dominating set of the r-th neighborhood of a vertex v, $VED(N_G^r[v])$ with respect to G as follows. Find a maximal independent set I for the graph induced by the vertices of $N_G^r[v]$. Observe that if we choose each vertex $v_i \in I$ in $VED(N_G^r[v])$, then it forms a vertex-edge dominating set for $N_G^r[v]$ (see Lemma 5.3.3).



$N_G[u] \notin VED(N_G^r[v])$ $N_G[v] \notin VED(N_G^r[v])$

Figure 5.7: $N_G[u], N_G[v] \notin I$

Lemma 5.3.3. *$VED(N_G^r[v])$ is a VEDS of $N_G^r[v]$ in G.*

Proof. Suppose to the contrary, assume that $VED(N_G^r[v])$ is not a VEDS of the graph $G' = (V', E')$ induced by $N_G^r[v]$. That means, there exist an edge $uv \in E'$ such that $N_{G'}[u] \notin VED(N_G^r[v])$ and $N_{G'}[v] \notin VED(N_G^r[v])$ (see Figure 5.7). It contradicts the fact that I is a maximal independent set in G'. Thus, the lemma. □

Lemma 5.3.4. *The maximum size of a vertex-edge dominating set of the r-th neighborhood of a vertex v is bounded by $(r+2)^2$, i.e.,$|VED(N_G^r[v])| \leq (r+2)^2$.*

Proof. We compute a maximal independent set I before computing a vertex-edge dominating set in the graph $G' = (V', E')$ induced by $N_G^r[v]$. The cardinality of a maximal independent set in the UDG G' is bounded by the number of non-intersecting unit disks packed in a disk of radius $r+2$ centered at v. So, $|I| \leq \frac{\pi(r+2)^2}{\pi(1)^2} = (r+2)^2$. From Lemma 5.3.3, in any graph, the cardinality of a minimum vertex-edge dominating set is bounded by the cardinality of maximal independent set. Therefore, $|VED(N_G^r[v])| \leq (r+2)^2$. □

Lemma 5.3.5. *For $\rho = 1+\epsilon$, there always exists an r violating the condition $|(VED(N_G^{r+4}[v])| > \rho|VED(N_G^r[v])|$.*

Proof. On the contrary, suppose that there exist a vertex $v \in V$ such that
$|(VED(N_G^{r+4}[v])| > \rho|VED(N_G^r[v])|$ for all $r = 1, 2, \ldots$.
From Lemma 5.3.4, we have $|(VED(N_G^r[v])| \leq (r+2)^2$.

Therefore, if $r = 4k$,

$(r+2)^2 \geq |(VED(N_G^r[v])| > \rho|VED(N_G^{r-4}[v])| > \cdots > \rho^{\frac{r}{4}}|VED(N_G^2[v])| \geq \rho^{\frac{r}{4}}$,

and if $r = 4k + s$ for $1 \leq s \leq 3$,

$(r+2)^2 \geq |(VED(N_G^r[v])| > \rho|VED(N_G^{r-4}[v])| > \cdots > \rho^{\frac{r-1}{4}}|VED(N_G^1[v])| \geq \rho^{\frac{r-1}{4}}$.

Hence,

$$(r+2)^8 > \begin{cases} \rho^r, & \text{if } r \text{ is } 4k \\ \rho^{r-1}, & \text{if } r \text{ is } 4k+s \end{cases} \tag{5.1}$$

Observe that in the inequality (5.1), the right side is an exponential function where as the left side is a polynomial function in r, which results in a contradiction for $\rho > 1$. □

Lemma 5.3.6. *The smallest r violating inequality (5.1) is bounded by $O(\frac{1}{\epsilon}\log\frac{1}{\epsilon})$.*

Proof. Let $\hat{r}$ be the smallest r violating the inequalities (5.1). We prove $\hat{r} \leq O(\frac{1}{\epsilon}\log\frac{1}{\epsilon})$ by using the inequalities (i) $\log(1+\epsilon) > \frac{\epsilon}{2}$ for $0 < \epsilon < 1$, (ii) $\log x < x$ for $x > 1$, and (iii) $\log\frac{1}{\epsilon} \geq 1$ for $\epsilon \leq \frac{1}{10}$. Let $x = \frac{c}{\epsilon}\log\frac{1}{\epsilon}$. Consider the inequality $(x+2)^8 \leq (2x)^8 \leq (1+\epsilon)^x$. The former inequality trivially holds for the choice of x and any $\epsilon > 0$, and taking the logarithm on both sides of the latter inequality, we get $\frac{8\log 2x}{x} \leq \log(1+\epsilon)$. By inequality (i), now, it suffice to show that $\frac{8\log 2x}{x} \leq \frac{\epsilon}{2}$. Using the inequalities (ii) and (iii), the choice of x satisfies the inequality for any constant c satisfying $\log 2c < \frac{c}{16}$. □

Lemma 5.3.7. *For a given $v \in V$, minimum vertex-edge dominating set $VED_{opt}(\mathcal{R}_i)$ of $\mathcal{R}_i$ can be computed in polynomial time.*

Proof. Let $G' = (V', E')$ be a graph induced by $\mathcal{R}_i \subseteq N_G^{r+4}[v]$. From Lemma 5.3.4, the size of $N_G^{r+4}[v]$ is bounded by $O(r^2)$, so we take every possible tuple of size at most $O(r^2)$ and check whether the selected tuple is a vertex-edge dominating set of the graph G'. This process takes $O(\binom{n}{r^2}) = O(n^{r^2})$ time. Since $r = O(\frac{1}{\epsilon}\log\frac{1}{\epsilon})$ by Lemma 5.3.6, $VED_{opt}(\mathcal{R}_i)$ can be computed in polynomial time. □

Lemma 5.3.8. *For the collection of subsets $\{\mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_k\}$, $\mathcal{D} = \bigcup_{i=1}^{k} VED(\mathcal{R}_i)$ is a vertex-edge dominating set in $G = (V, E)$.*

Proof. To prove $\mathcal{D}$ is a vertex-edge dominating set of the graph G, we need to prove for every edge $v_iv_j \in E$, there exist at least one vertex from $N_G[v_i]$ or $N_G[v_j]$ in $\mathcal{D}$. It follows from our construction of the subsets $\mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_k$ (Section 5.3.1) that each edge v_iv_j belongs to a particular subset $\mathcal{R}_i$ and $VED(\mathcal{R}_i)$ is a vertex-edge dominating set of the graph induced by the vertices of $\mathcal{R}_i$. Thus the lemma. □

Corollary 5.3.9. $\mathcal{D}^* = \bigcup_{i=1}^{k} VED_{opt}(\mathcal{R}_i)$ *is a vertex-edge dominating set in* G, *for the collection* $\mathcal{R} = \{\mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_k\}$.

Theorem 5.3.10. *For a given UDG,* $G = (V, E)$, *and an* $\epsilon > 0$, *we can design a* $(1+\epsilon)$-*factor approximation algorithm for VEDS problem in G with running time* $n^{O(c^2)}$, *where* $c = O(\frac{1}{\epsilon}\log\frac{1}{\epsilon})$.

Proof. Follows from Lemmata 5.3.2, 5.3.5, 5.3.7, and Corollary 5.3.9. □

5.4 Conclusion

In this chapter, we studied the geometric minimum vertex-edge dominating set (GMVEDS) problem in unit disk graphs and showed that the GMVEDS problem belongs to the NP-hard class. We also proposed a simple polynomial-time 4-factor approximation algorithm and a PTAS for the GMVEDS problem.

Chapter 6

Total Dominating Set Problem in Unit Disk Graphs

In this chapter, we study the geometric version of the minimum total dominating set (GMTDS) problem, defined as follows:

Definition 6.0.1. (Total domination) *Given a simple unit disk graph* $G = (V, E)$ *corresponding to a point set* $P = \{p_1, p_2, \ldots, p_n\}$ *for disk centers in the plane, find a minimum cardinality subset* $D_t \subseteq V$ *such that for each vertex* $p_i \in V$, $|D_t \cap N_G(p_i)| \geq 1$, *where* $N_G(p_i)$ *is known as the open neighborhood of the vertex* $p_i \in V$.

In a simple undirected graph $G = (V, E)$, the *open neighborhood* of a vertex $v \in V$ is the set $N_G(v) = \{u \in V : uv \in E\}$.

Observe that, a vertex $v \in D$ (dominating set) dominates all its neighbors and itself whereas a vertex $v \in D_t$ (total dominating set) dominates all its neighbors other than itself.

The goal of this chapter is (i) to prove that the problem belongs to the NP-hard class, (ii) to propose a simple 8-factor approximation algorithm, and (iii) to propose a PTAS for the problem.

6.1 NP-hardness Result of the GMTDS Problem

In this section, we show that the decision version of the TDS problem in UDGs belongs to the class NP-complete. The *vertex cover* (VC) problem in the planar graph with maximum degree 3 is known to be in NP-hard class [43]. To prove the NP-hardness result of the TDS problem in UDGs, we use polynomial-time reduction from the vertex cover problem in the planar graph to it. Now, we define the decision version of the TDS problem in UDGs and vertex cover problem in planar graphs as follows:

Definition 6.1.1. (The TDS problem in UDGs (Tds-Udg)) *Given a unit disk graph G and an integer $k(> 0)$, does there exist a TDS of size at most k?*

Definition 6.1.2. (The VC problem in planar graphs (Vc-Pla)) *Given a planar graph G with maximum degree 3 and an integer $k(> 0)$, does G has a VC of size at most k?*

Corollary 6.1.3. *Let $G = (V, E)$ be a planar graph with maximum degree 3. The graph G can be embedded in linear time on a planar grid of size $4n \times 4n$ using $O(|V|^2)$ area such that the coordinate of each vertex $v \in V$ is $(4i, 4j)$ for some integers i, j and each edge $e \in E$ is a finite sequence of consecutive line segments of length 4 units along the grid lines.*

Proof. Follows from the Lemma 3.1.4. □

Lemma 6.1.4. *For a given* Vc-Pla *instance $G = (V, E)$ with at least one edge, an instance $G' = (V', E')$ of* Tds-Udg *can be constructed in polynomial-time.*

Proof. Let $V = \{v_1, v_2, \dots, v_n\}$ and $E = \{e_1, e_2, \dots, e_m\}$. The construction of G' from the graph G is described in the following four steps.

Step 1: (Embedding) We first embed G into a planar grid of size 4×4 using Corollary 6.1.3. On the embedding, each of the vertex $v_i \in V$ becomes grid point p_i and each edge $e_j \in E$ become a finite sequence of connected line segment(s) of length four units

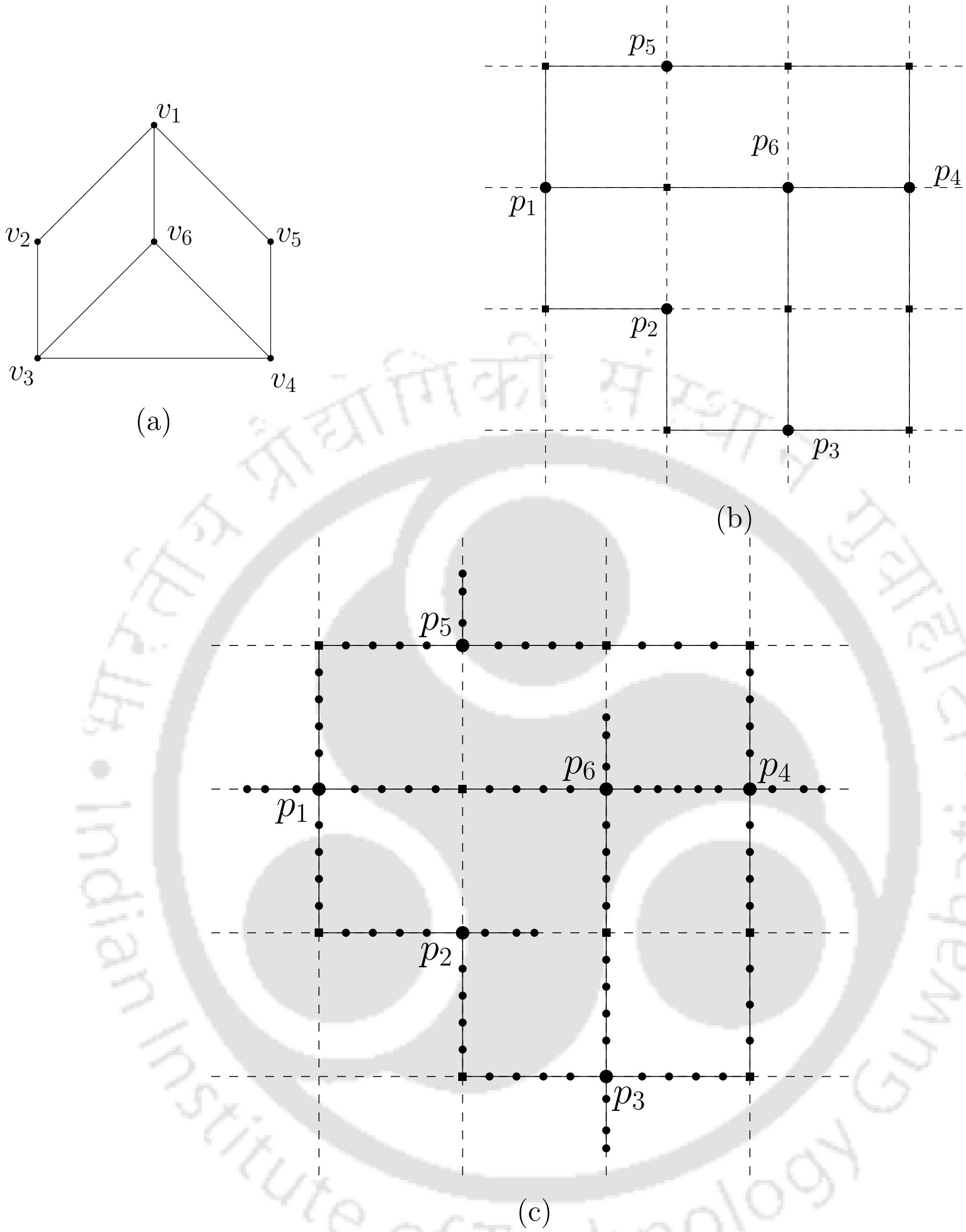


Figure 6.1: (a) A planar graph G with maximum degree 3, (b) its embedding on a 4×4 grid, and (c) construction of an UDG from the embedding.

along the grid lines. Assume that ℓ is the total number of line segments used in the embedding. We call the point p_i corresponding to the vertex $v_i \in V$ $(i = 1, 2, \ldots, n)$ in the embedding as *vertex points* (see Figure 6.1(a) and 6.1(b)). Let N be the set of vertex points. Therefore, $N = \{p_i \mid v_i \in V\}$ and $|N| = |V|(= n)$.

Step 2: (Extra points) In this step, we add some extra points on each of the ℓ line segments (obtained in embedding step) so that unit disks centered on these points and grid points (see embedding step) form a unit disk graph as follows: (a) for each edge p_ip_j with only one line segment i.e., length of the edge is 4 units, we add five points at distance 0.98, 1.49, 2, 2.51, 3.02 units from p_i (see edge p_4p_6 in Figure 6.1(c)), and (b) for each edge p_ip_j with more than one segment i.e., length of the edge is greater than 4 units, (i) add a point on each of the grid point on the edge other than the vertex point and name it as grid point (see square points in Figure 6.1(c)), and (ii) we add four points on each of the line segments connected with p_i and p_j at distances 1, 1.75, 2.5, 3.25 units from p_i and p_j, and for other line segments we add three points at distance 1 units from each other excluding the *grid points* (see the edge p_3p_4 in Figure 6.1(c)). Let A be the set of all points added in this step. Therefore, $|A| = 4\ell + m$, where ℓ is the total number of grid line segments in the embedding.

Step 3: (Support point) Add a new line segment of length 1.4 units at each of the vertex points p_i without coinciding with the line segments that had already been drawn in the embedding. Observe that the addition of such a line segment is possible without losing the planarity as the maximum degree of G is 3. We add three points x_i, y_i, z_i on each of these line segments at distances 0.3, 1.1, and 1.4 units from the corresponding vertex point p_i. Let S be the set of all points added in this step. Therefore, $|S| = 3n$.

Step 4: (Construction of UDG) We construct a UDG $G' = (V', E')$, where $V' = N \cup A \cup S$ and $E' = \{u'v' : u', v' \in V'$ and the Euclidean distance between u' and v' is at most 1 unit$\}$ (see Figure 6.1(c)). From Lemma 6.1.3, $\ell = O(n^2)$. Therefore both $|V'|$ and $|E'|$ are bounded by $O(n^2)$. Hence, G' can be constructed in polynomial-time. □

Theorem 6.1.5. Tds-Udg *belongs to the class NP-complete.*

Proof. Let $T \subseteq V$ be an arbitrary subset of vertices and $k(> 0)$ be an integer. Observe that, we can verify whether T is a total dominating set such that $|T| \leq k$ or not in polynomial-time. Therefore, Tds-Udg $\in$ NP.

To prove the NP-hardness of TDS-UDG, we will use polynomial time reduction of VC-PLA to it. We construct an instance $G' = (V', E')$ of TDS-UDG from an arbitrary instance $G = (V, E)$ of VC-PLA in polynomial time using the steps mentioned in Lemma 6.1.4. Next, we prove the following claim to complete the NP-hardness proof of TDS-UDG.

Claim: *G has a vertex cover C with $|C| \leq k$ if and only if G' has a total dominating set T with $T \leq k + 2\ell + 2n$.*

Necessity: Let $C \subseteq V$ be a vertex cover of G such that $|C| \leq k$. Let $N' = \{p_i \in N \mid v_i \in C\}$, i.e., N' is the set of vertices (or vertex points) in G' that correspond to the vertices in C. From each segment, we choose 2 vertices (extra points) from A and corresponding to each vertex point, we choose 2 points (support points) from S, in the embedding. The set of chosen vertices, say $A'(\subseteq A)$, $S'(\subseteq S)$, together with N' will form a TDS of desired cardinality in G'. We now discuss the process of obtaining the set A'. Initially $A' = \emptyset$. As C is a vertex cover, every edge in G has at least one of its end vertices in C. Let v_iv_j be an edge in G and $v_i \in C$ (choose any of them arbitrarily if both v_i and v_j are in C). Note that the edge v_iv_j is represented as a sequence of line segments in the embedding. Start traversing the segments (of v_iv_j) from p_i, where p_i corresponds to v_i, and add two consecutive vertices by leaving two consecutive vertices in between starting from p_i to A' in the traversal (see p_4p_5 in Figure 6.2(b)). The red bold vertices are part of A' while traversing from p_4).

Apply the above process to each edge in G. Observe that the cardinality of A' is 2ℓ as we have chosen 2 vertices from each segment in the embedding. Next, we choose $2n$ points from S in $S' = \{x_i, y_i : p_i \in N\}$. Let $T = N' \cup A' \cup S'$. Now, we argue that T is a total dominating set in G'.

For each point $p_i \in N$, p_i is dominated by x_i, x_i is dominated by y_i, y_i is dominated by x_i and z_i is dominated by y_i. So, the sets N and S satisfies total domination condition. Now it is remaining to prove that the set A satisfies total domination condition. Observe the way we have chosen points from A in T, with a gap of two consecutive points, two consecutive points are chosen in T. For each point $p_i \in T$, p_i dominates $N_G(p_i) \in A$ and

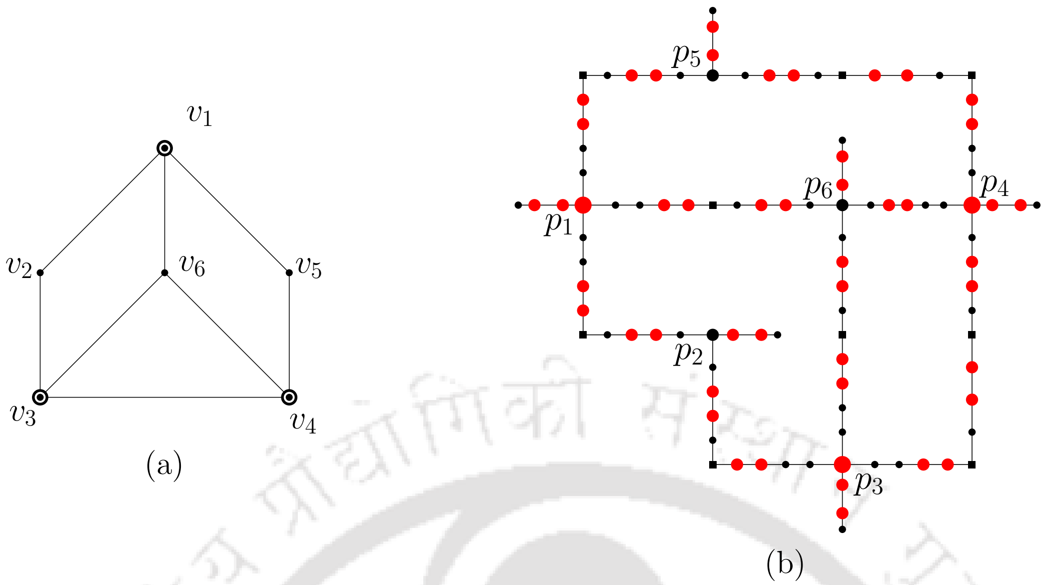


Figure 6.2: (a) A vertex cover $\{v_1, v_3, v_4\}$ in G, and (b) the construction of A' in G' (the tie between v_3 and v_4 is broken by choosing v_3)

the selected points of A in T can total dominate all the remaining points of A (see the edge p_1p_2 in Figure 6.2(b)).

Therefore, T is a TDS in G' and $|T| = |N'| + |A'| + |S'| \leq k + 2\ell + 2n$.

Sufficiency: Let $T \subseteq V'$ be a TDS of size at most $k + 2\ell + 2n$. We prove that G has a vertex cover of size at most k with the help of the following claims.

(i) Out of three support points associated with each $p_i \in N$, at least two points belongs to T, i.e., $|S \cap T| \geq 2n$.

(ii) Every segment in the embedding must contribute at least two points to T and hence $|A \cap T| \geq 2\ell$, where ℓ is the total number of segments in the embedding.

(iii) If p_i and p_j correspond to end vertices of an edge v_iv_j in G, and if both p_i, p_j are not in T, then there must be at least $2\ell' + 1$ vertices in T from the segment(s) representing the edge v_iv_j, where ℓ' is the number of segments representing the edge v_iv_j in the embedding.

Claim (i) directly follows from the definition of total dominating set. Observe that we added points x_i, y_i, z_i such that p_i is adjacent to x_i, x_i is adjacent to y_i, and y_i is adjacent to z_i in G', i.e., $\{p_i x_i, x_i y_i, y_i z_i\} \subseteq E'$ for each i. Hence, y_i must be in T as y_i is the only vertex that can dominate z_i and either x_i or z_i must be in T to dominate y_i. Therefore, any total dominating set of G' must contain two support points in T out of three support points associated with $p_i, 1 \leq i \leq n$, i.e., $|S \cap T| \geq 2n$.

Claim (ii) follows from the fact that only consecutive points are adjacent (in G') on any segment in the embedding. Let η be a segment in the embedding having vertices q_i, q_{i+1}, q_{i+2}, and q_{i+3}. On contrary, assume that η has only one of its vertices in T. Note that only q_i can not be in T. If q_i present in T, then q_{i+2} is not dominated by any point, which is a contradiction to the fact that T is a TDS. If q_{i+1} is the only point in T then q_{i+1} is not dominated by any point. If q_{i+2} will be chosen as the only point from η in T then q_{i+2} is not dominated by any other point and finally if q_{i+3} will be chosen then q_{i+1} is not dominated by any other point. In all cases, we arrived at a contradiction.

Claim (iii) follows from the definition of the total dominating set that any point chosen in the solution set dominates all its neighbors other than itself. Here, any point selected from a segment in T has exactly two neighbors other than itself. So, it can dominate at most 2 points. There are ℓ' segments between two node points p_i and p_j having $4\ell' + 1$ number of points and both p_i and p_j are not in T. So, the minimum number of points required in T to ensure total domination is $\lceil \frac{4\ell'+1}{2} \rceil = 2\ell' + 1$.

Now, we will show that, by removing and/or replacing some vertices in T, a set of at most k points from N can be chosen such that the corresponding vertices form a vertex cover in G. The vertices in S account for $2n$ vertices in T (due to Claim (i)). Let $T = T \setminus S$ and $C = \{v_i \in V \mid p_i \in T \cap N\}$. If any edge $v_i v_j$ in G has none of its end vertices in C, then we do the following:

consider the sequence of segments representing the edge $v_i v_j$ in the embedding. Since, both p_i and p_j are not in T, there must exist a segment having three vertices in T (due to Claim (iii)). Consider the segment having its three vertices in T. Delete any one of the vertices on the segment and introduce p_i (or p_j). Update C and repeat the process till every edge has at least one of its end vertices in C. Due to Claim (ii), C is a vertex cover in G with $|C| \leq k$. Therefore, TDS-UDG $\in$ NP-hard. As TDS-UDG $\in$ NP and

TDS-UDG $\in$ NP-hard, TDS-UDG $\in$ NP-complete. □

6.2 Approximation Algorithm for GMTDS Problem

In this section, we propose a simple 8-factor approximation algorithm for the TDS problem in UDGs. The worst-case time complexity of our proposed algorithm is $O(n \log k)$, where k is the output size of our algorithm. Note that, if there exist an r-factor approximation algorithm for the dominating set problem then a $2r$-factor approximation algorithm for the total dominating set problem can be obtained by adding one neighbor for each vertex in the dominating set. A 10-factor approximation algorithm is available in the literature for the TDS problem in UDGs [70]. In this section, we improved the approximation factor of the TDS problem in UDGs to 8.

Lemma 6.2.1. *[70] Let C be a unit disk centered at a point p. If S is a set of independent[1] unit disks such that every unit disk in S contains the point p then $|S| \leq 5$.*

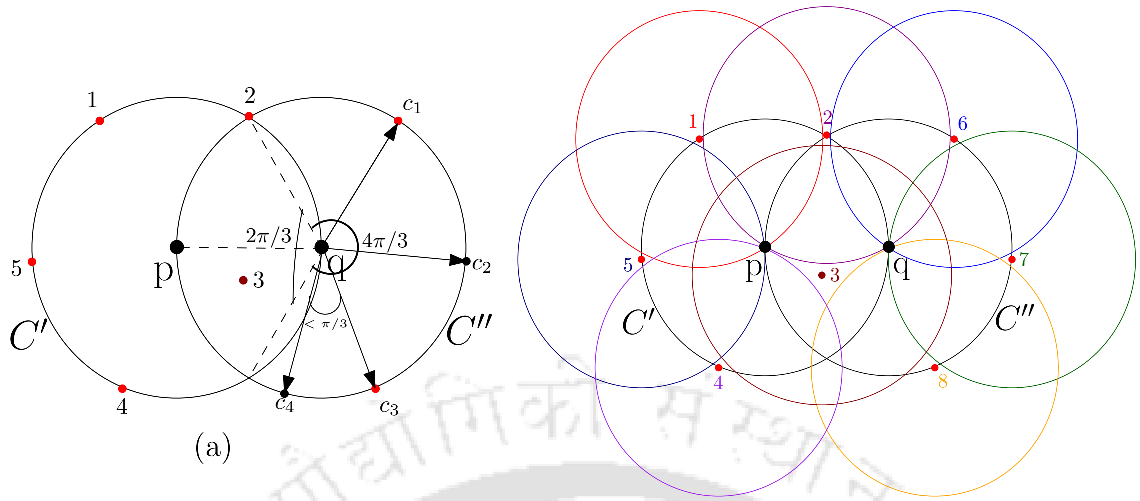
Lemma 6.2.2. *Let C' and C'' be two unit disks centered at points p and q respectively, such that p and q are at one unit distance apart. If S is a set of independent unit disks containing the point p and/or q, then $|S| \leq 8$.*

Proof. Let S_p, $S_q \subseteq S$ be the sets of independent unit disks containing the point p and q respectively. This implies $S = S_p \cup S_q$. Note that, the cardinality of S will increase if the distance between p and q increases. Therefore, without loss of generality, we assume that the points p and q are one unit distance apart, i.e., the point q lies on the boundary of the unit disk C' centered at p (see Figure 6.3 for reference). Now, we consider the following two cases:

Case (i): $|S_p| \leq 3$. From Lemma 6.2.1, we have $|S_q| \leq 5$. Hence, $|S_p \cup S_q| \leq 8$.

Case (ii): $|S_p| > 3$. It is known that $|S_p| \leq 5$ (see Lemma 6.2.1). So $|S_p|$ is either 4 or 5.

[1] Two disks are independent if the distance between their centers is greater than one.



(a) (b)

Figure 6.3: Illustration of Lemma 6.2.2.

(a) Assume that $|S_p| = 5$. Now it is remaining to prove that, there does not exist more than 3 independent unit disks in the set (say S'_q) $\subseteq S_q$, which contain the point q but does not contain the point p, i.e., $|S'_q| = |S_q| - |S_p \cap S_q| \leq 3$. For the sake of argument assume that, $|S'_q| = 4$, which means, there exist 4 unit disks centered at c_i, $1 \leq i \leq 4$ in S'_q, which contain the point q. Let us denote the ray $\overrightarrow{qc_i}$ by r_i, $(1 \leq i \leq 4)$. Since there are 4 rays coming out from q and the disks centered at c_i, $(i = 1, 2, 3, 4)$ does not contains p, there exist one pair of rays r_i and r_j such that the angle between them is at most $\pi/3$ (see Figure 6.3(a)), which leads to a contradiction that the disks centered at c_i, $(i = 1, 2, 3, 4)$ are independent. Thus, $|S'_q| \leq 3$, which implies $|S_p \cup S_q| \leq 8$ (see Figure 6.3(b) for illustration).

(b) Assume that $|S_p| = 4$. To prove $|S_p \cup S_q| \leq 8$, it is sufficient to prove that $|S'_q| < 5$. For the sake of argument assume that, $|S'_q| = 5$, that means, there exist 5 independent unit disks in the set S'_q, which contain the point q but does not contain the point p. Following a similar argument as in case (a), it leads to a contradiction. Thus, $|S'_q| < 5$, which implies $|S_p \cup S_q| \leq 8$.

In either of the cases, we proved that $|S_p \cup S_q| \leq 8$. Therefore, $|S| \leq 8$. □

Let P denote the set of n points (center of the disks) given in the plane $\mathbb{R}^2$. We use

$\Delta(S)$ to denote the unit disks centered at the points in a subset $S \subseteq P$. A dominating set in unit disk graphs (Ds-Udg) $D \subseteq P$ of the set of disks $\Delta(P)$ is said to be an independent Ds-Udg if for each pair p, $q \in D$, $p \notin N_G(q)$.

The procedure of generating a Tds-Udg for a given points set P in $\mathbb{R}^2$ is described in Algorithm 6.1.

Algorithm 6.1 Total dominating set in P

Require: A set of disks $\Delta(P)$
Ensure: A total dominating set T of $\Delta(P)$

```
1:  D ← ∅, and T ← ∅
2:  while (P ≠ ∅) do
3:      choose an arbitrary point p ∈ P
4:      D ← D ∪ {p}; P ← P \ N_G[p]          ▷ N_G[p] = N_G(v) ∪ {v}
5:  end while
6:  for every p ∈ D do
7:      if N_G(p) ∩ T = ∅ then
8:          let q ∈ N_G(p)
9:          T = T ∪ {q}
10:     end if
11: end for
12: T = T ∪ D
13: return T
```

Lemma 6.2.3. *T returned by Algorithm 6.1 is a* Tds-Udg *for the set of unit disks in $\Delta(P)$.*

Proof. Initially, Algorithm 6.1 find an independent Ds-Udg D (see **while** loop in line number 2 of the algorithm), which ensures domination for the set of unit disks $\Delta(P)$ and total domination for the points $P \setminus D$. Next, to obtain total domination in D, for each point $p \in D$ the algorithm ensures the existence of a point $q \in N_G(p)$ in T (see **for** loop in line number 5 of the algorithm). The selected points in T along with D ensures total domination for the set of unit disks in $\Delta(P)$. □

Lemma 6.2.4. *$|T| \leq 8|OPT|$, where OPT is a* Tds-Udg *for the unit disks $\Delta(P)$ of minimum size.*

Proof. Consider an arbitrary point $p \in OPT$. As OPT is a TDS of minimum size there must exist a point $q \in OPT$ such that $p \in N_G(q)$ and the point q ensures the domination for the point p. Now consider both the unit disks centered at p and q. From Lemma 6.2.2, it is proved that there exist at most 8 unit disks in an independent Ds-Udg D of $\Delta(P)$ that can contain the points p and/or q.

Now, the cardinality of T follows from the fact that for every pair $p, q \in OPT$ such that $p \in N_G(q)$, there may exist at most 8 unit disks in an independent Ds-Udg D of $\Delta(P)$ that can contain the points p and/or q. Our algorithm chooses at most one neighbor for each point in D in to the solution set T, to make T as a Tds-Udg (see line numbers 6-12 of Algorithm 6.1). So for 8 points in D at most 16 points may be chosen in T with respect to two points $p, q \in OPT$. which leads at most 16 points are chosen by Algorithm 6.1 against 2 points (namely, p and q) chosen in the optimal solution of the TDS, i.e., $|T| \leq \frac{16}{2}|OPT|$. Therefore, $|T| \leq 8|OPT|$. □

Lemma 6.2.5. *The worst case time required to generate a* Tds-Udg *for the set of disks* $\Delta(P)$ *by Algorithm 6.1 is* $O(n \log k)$, *where* k *is the size of the output.*

Proof. We now describe the time complexity of Algorithm 6.1 for computing a Tds-Udg T of $\Delta(P)$ as follows:

Let us assume that, $\mathcal{R}$ is an axis parallel rectangular region containing the points in P. We partition $\mathcal{R}$ into grid cells of size 1×1. A point $p_i = (x_i, y_i) \in P$ lies in the grid cell indexed by $[\lfloor x_i \rfloor, \lfloor y_i \rfloor]$ for $i = 1, 2, \ldots, n$. Each grid cell is attached with a list of points in P that are belong to that cell. We construct an independent dominating set D for UDG corresponding to $\Delta(P)$. While considering a point $p_i \in P$, we inspect all members of D which are attached to all 9 cells $[\alpha, \beta]$, where $\lfloor x_i \rfloor - 1 \leq \alpha \leq \lfloor x_i \rfloor + 1$ and $\lfloor y_i \rfloor - 1 \leq \beta \leq \lfloor y_i \rfloor + 1$. If there does not exist any unit disk d in D that contains the point p_i, we add p_i in D. Observe that, at the end of considering all the points in P, D will be an independent Ds-Udg for the set of disks in $\Delta(P)$. Initially, take $T = \emptyset$. Now, for each point $p \in D$, if there does not exist any point q in T such that $q \in N_G(p)$, then add q in T, and the existence of q is guaranteed otherwise finding TDS for this given instance is impossible as the point p is an isolated point. Finally, update $T = T \cup D$. After ensuring the existence of a point $q \in N_G(p)$ for each point $p \in D$, T is a TDS-UDG for the set of disks in $\Delta(P)$. Note that, (i) a grid cell may contain at most 6 points in T, and (ii) the number of grid cells to be inspected while processing a point $p_i \in P$ is at most 9. We use a height-balanced binary tree to store the indices of the grid cells containing a non-zero number of points in T. Thus, the time complexity for processing a point $p \in P$ is $O(\log k)$, where $k = |T|$ and $|P| = n$. □

Theorem 6.2.6. *Algorithm 6.1 is an 8-factor approximation algorithm for the* TDS-UDG *problem. The running time of the algorithm is $O(n \log k)$, where n is the input size and k is the output size.*

Proof. Follows from Lemma 6.2.4 and Lemma 6.2.5. □

6.3 Approximation Scheme for GMTDS Problem

In this section, we propose a polynomial-time approximation scheme (PTAS) for the TDS problem in unit disk graphs. We use the shifting strategy [55] technique to propose a PTAS. Let P be a point set (centers of the disks) given in a rectangular region $\mathcal{R}$ along with a fixed integer $k \geq 1$.

We use a two-level nested shifting strategy to propose a PTAS for the said problem. The first level of shifting strategy applied in the horizontal direction on $\mathcal{R}$. There are k iterations in the first level and the i-th iteration $(1 \leq i \leq k)$ partition the region $\mathcal{R}$ into many horizontal strips, where the first strip is of width $2i$, and remaining strips other than the last strip are of width $2k$. The width of the last strip may be less than $2k$. Without loss of generality, assume that each point lying on the left boundary of a strip belongs to its left adjacent strip. Now consider all the non-empty horizontal strip H, and apply the second level of shifting strategy on the vertical direction. In the second level of shifting strategy, the j-th iteration $(1 \leq j \leq k)$ partition each non-empty horizontal strip H into square/rectangular cells of size $2j \times \ell$ for the first cell and $2k \times \ell$ for all other cells, where ℓ defines the width of the strip H ($\ell = 2i$ for the first strip and $\ell = 2k$ for all other strips except the last strip).

We consider each non-empty $2k \times 2k$ squares (conceptually extending the smaller cells into $2k \times 2k$ square) and find the optimal solution of each square. The union of the optimal solution of each $2k \times 2k$ squares gives a feasible solution of each strip H. Finally, we take the union of the solution of each non-empty horizontal strip to get a feasible solution of the problem in a single iteration, i.e., (i, j)-th iteration. In the same process, we get the feasible solutions of all the iterations in the first level. We report the solution T, having minimum cardinality among all the solutions generated in each iteration as the solution of the TDS-UDG problem.

Now, we discuss the procedure of getting an optimal solution from each $2k \times 2k$ square. We first partition the cell of size $2k \times 2k$ into $(\lceil 2\sqrt{2}k \rceil)^2$ sub-cells. The size of each sub-cell is $\frac{1}{\sqrt{2}} \times \frac{1}{\sqrt{2}}$. Observe that, choosing any two points inside a sub-cell of size $\frac{1}{\sqrt{2}} \times \frac{1}{\sqrt{2}}$ ensures total domination for all unit disk centered in that sub-cell. Hence, the maximum number of points required to ensure total domination in a square of size $2k \times 2k$ is at most $2(\lceil 2\sqrt{2}k \rceil)^2$. Therefore, we have to check all possible combinations of points up to $2(\lceil 2\sqrt{2}k \rceil)^2$ to get an optimal solution in a cell χ of size $2k \times 2k$. Note that, along with the points inside a cell χ, the points within 1 unit apart from χ is also play a crucial role to get an optimum solution of χ. Let n_χ be the number of points in P whose corresponding disks have a portion in the cell χ (n_χ includes the points inside χ along with the points within 1 unit apart from χ). Then, we have to choose at most $O(n_\chi^{2(\lceil 2\sqrt{2}k \rceil)^2})$ combinations of points for getting the optimum solution for the TDS-UDG problem in a cell χ of size $2k \times 2k$. Since the points in P centered in a cell is disjoint from that of the other cells, and a point in P can participate in computing the optimum solution of at most 9 cells, we have the following result.

Lemma 6.3.1. *The total time required for the (i, j)-th iteration of the algorithm is $O(n^{2(\lceil 2\sqrt{2}k \rceil)^2})$.*

Proof. The feasible solution of the (i, j)-th iteration is the union of the optimum solutions of all the cells constructed in that iteration. Finally, the algorithm returns the minimum among the k^2 feasible solutions corresponding to k^2 iterations. □

Theorem 6.3.2. *Given a set P of n points (center of the unit disks) in $\mathcal{R}$ and an integer $k \geq 1$, a total dominating set of size at most $(1 + \frac{1}{k})^2 \times |OPT|$ can be computed in $O(k^2 n^{2(\lceil 2\sqrt{2}k \rceil)^2)})$ time, where OPT is the optimum solution.*

Proof. Using the shifting strategy analysis given by Hochbaum and Maass [55], we analyze the approximation factor of our algorithm. Let OPT be an optimum solution for the TDS-UDG problem for the point set P, and $OPT' \subseteq OPT$ be such points chosen in OPT, which totally dominate some points outside the boundary of all the cells in an (i, j)-th iteration. Let T be a solution obtained by our algorithm in an iteration. Then, $|T| \leq |OPT| + |OPT'|$. For all the iterations of (i, j) $(1 \leq i, j \leq k)$, we have $\sum_{i=1}^{k} \sum_{j=1}^{k} |T| \leq k^2|OPT| + \sum_{i=1}^{k} \sum_{j=1}^{k} |OPT'|$. Since any point from a cell χ chosen in OPT can dominate points from no more than one horizontal strip (or vertical strip), and at most k times each horizontal (or vertical) boundary appears throughout the algorithm, we have $\sum_{i=1}^{k} \sum_{j=1}^{k} |OPT'| \leq k|OPT| + k|OPT|$. Thus, $\sum_{i=1}^{k} \sum_{j=1}^{k} |T| \leq k^2|OPT| + 2k|OPT| = (k^2 + 2k)|OPT|$. Therefore, $min \sum_{i=1}^{k} \sum_{j=1}^{k} |T| \leq (1 + \frac{1}{k})^2 \times |OPT|$. The time complexity result follows from Lemma 6.3.1. □

6.4 Conclusion

In this chapter, we considered the geometric minimum total dominating set (GMTDS) problem in unit disk graphs. We showed that the decision version of the GMTDS problem belongs to the class NP-complete. We proposed an almost linear time 8-factor approximation algorithm and a PTAS for the problem.

Chapter 7

Conclusions and Future Works

Most of the variants of dominating set problems are NP-hard in general graphs and they do not even admit constant factor approximation algorithms. This motivated researchers to study the dominating set and its variants in other graph classes. In this thesis, we have studied minimum dominating set problem and some of its variants in general graphs and unit disk graphs, namely, geometric maximum distance-d independent set (GMDdIS) problem, geometric minimum distance-d dominating set (GMDdDS) problem, minimum d-distance m-tuple (ℓ, r)-dominating set ((d, m, ℓ, r) set) problem, geometric minimum vertex-edge dominating set (GMVEDS) problem and geometric minimum total dominating set (GMTDS) problem. Unfortunately, these problems are NP-hard in unit disk graphs too, but unlike in general graphs, they admit constant factor approximation algorithms and approximation schemes, thus the search for faster approximation algorithms encouraged us to study these problems.

This chapter aims to explain the contribution made in this thesis, summarize it in a table, and state some of the problems which might be carried as future research.

We studied the DdIS problem, a variant and generalized version of the independent set problem, and DdDS problem, a generalized version of the dominating set problem, in unit disk graphs. We proved that both DdIS and DdDS problems belong to the NP-hard class in unit disk graphs. We proposed simple 4-factor approximation algorithms for both problems. We also proposed polynomial-time approximation schemes (PTAS) for both problems.

Next, we proved the 1-distance m-tuple (ℓ, r) domination problem belongs to the NP-hard class for each fixed value of m, ℓ, and r. We also proved the d-distance m-tuple $(\ell, 2)$ domination problem belongs to the NP-hard class for each fixed value of $d(>1), m$, and ℓ. We have showed that the first problem is not approximated within a factor of $(\frac{1}{2} - \varepsilon) \ln |V|$ for each fixed value of m, ℓ, and r, unless P = NP and the second problem is not approximated within a factor of $(\frac{1}{4} - \varepsilon) \ln |V|$ for each fixed value of $d(>1), m$, and ℓ, unless P = NP, where V is the vertex set of the input graph. The reduction in the NP-completeness/inapproximability proofs is very powerful as these are common reductions for completely different kinds of dominations based on the values of m, ℓ, and r.

Next, we studied the minimum vertex-edge dominating set (VEDS) problem in unit disk graphs and showed that the VEDS problem belongs to the NP-hard class in that class of graphs. We also proposed a polynomial-time 4-factor approximation algorithm and a PTAS for the VEDS problem.

Table 7.1: Summerizing Results obtained in the thesis

Problem	**NP-hardness**	**Approximation factor & Time complexity**	**PTAS & Time complexity**
GMDdIS	NP-hard in UDGs	4-factor in $d^2 n^{O(d)}$	$\frac{1}{(1+\frac{1}{k})^2} \times \|OPT\|$ in $k^2 n^{O(k)}$
GMDdDS	NP-hard in UDGs	4-factor in $d^2 n^{O(d)}$	$(1+\frac{1}{k})^2 \times \|OPT\|$ in $k^2 n^{O(k)}$
$((1, m, \ell, r)$ set)	NP-hard in Graphs	inapproximable in $(\frac{1}{2} - \varepsilon) \ln \|V\|$	NA
$((d, m, \ell, 2)$ set)	NP-hard in Graphs	inapproximable in $(\frac{1}{4} - \varepsilon) \ln \|V\|$	NA
GMVEDS	NP-hard in UDGs	4-factor in n^{30}	$(1-\epsilon) \times \|OPT\|$ in $O(n^{O(\frac{1}{\epsilon} \log \frac{1}{\epsilon})})$
GMTDS	NP-hard in UDGs	8-factor in $O(n \log k)$	$(1+\frac{1}{k})^2 \times \|OPT\|$ in $O(k^2 n^{2(\lceil 2\sqrt{2}k \rceil)^2)})$

Finally, we considered the minimum total dominating set (TDS) problem in unit disk graphs. We showed that the TDS problem belongs to the NP-hard class in unit disk graphs. We proposed an almost linear time 8-factor approximation algorithm and a PTAS for the same problem.

As part of future work, we would like to introduce the studied problems in this thesis in other intersection graphs of disks of arbitrary radius, rectangles, convex polygons, etc. In general, it is hard to approximate many variants of dominating set problems within a constant-factor in general graphs. We would like to consider those problems in different graph classes and try to propose constant-factor approximation algorithms and PTASes for them. We would like to improve the existing constant-factor results for many other variants of dominating set problems by considering them in UDGs as well as in many other geometric intersection graphs. It would be interesting to see the results for the d-distance m-tuple (ℓ, r)-domination problem in different graph classes as well as in UDGs too.

Bibliography

[1] G. Agnarsson, P. Damaschke, and M. M. Halldórsson. Powers of geometric intersection graphs and dispersion algorithms. *Discrete Applied Mathematics*, 132(1):3–16, 2003.

[2] A. Alimadadi, M. Chellali, and D. A. Mojdeh. Liar's dominating sets in graphs. *Discrete Applied Mathematics,*, 211:204–210, 2016.

[3] K. M. Alzoubi, P. Wan, and O. Frieder. Message-optimal connected dominating sets in mobile ad hoc networks. In *Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing*, pages 157–164, 2002.

[4] D. Amos and E. DeLaVina. On total domination in graphs. *Senior Project, University of Houston Downtown*, 2012.

[5] D. V. Andrade, M. GC Resende, and R. F. Werneck. Fast local search for the maximum independent set problem. *Journal of Heuristics*, 18(4):525–547, 2012.

[6] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and the hardness of approximation problems. *Journal of the ACM (JACM)*, 45(3):501–555, 1998.

[7] T. Biedl and G. Kant. A better heuristic for orthogonal graph drawings. *Computational Geometry*, 9(3):159–180, 1998.

[8] A. Biswas, P. Bhowmick, and B. B. Bhattacharya. Tips: On finding a tight

isothetic polygonal shape covering a 2d object. In *Scandinavian Conference on Image Analysis*, pages 930–939. Springer, 2005.

[9] A. Biswas, P. Bhowmick, M. Sarkar, and B. B. Bhattacharya. A linear-time combinatorial algorithm to find the orthogonal hull of an object on the digital plane. *Information Sciences*, 216:176–195, 2012.

[10] K. Bose, M. K. Kundu, R. Adhikary, and B. Sau. Distributed localization of wireless sensor network using communication wheel. In *International Symposium on Algorithms and Experiments for Sensor Systems, Wireless Networks and Distributed Robotics*, pages 17–31. Springer, 2020.

[11] R. Boutrig and M. Chellali. Total vertex-edge domination. *International Journal of Computer Mathematics*, 95(9):1820–1828, 2018.

[12] R. Boutrig, M. Chellali, T. W. Haynes, and S. T. Hedetniemi. Vertex-edge domination in graphs. *Aequationes mathematicae*, 90(2):355–366, 2016.

[13] R. C. Brigham, J. R. Carrington, and R. P. Vitray. Connected graphs with maximum total domination number. *Journal of Combinatorial Mathematics and Combinatorial Computing*, 34:81–96, 2000.

[14] S. Butenko, P. Pardalos, I. Sergienko, V. Shylo, and P. Stetsyuk. Finding maximum independent sets in graphs arising from coding theory. In *Proceedings of the 2002 ACM symposium on Applied computing*, pages 542–546, 2002.

[15] P. Carmi, G. K. Das, R. K. Jallu, S. C. Nandy, P. R. Prasad, and Y. Stein. Minimum dominating set problem for unit disks revisited. *International Journal of Computational Geometry & Applications*, 25(3):227, 2015.

[16] P. Carmi, M. J. Katz, and N. Lev-Tov. Polynomial-time approximation schemes for piercing and covering with applications in wireless networks. *Computational Geometry*, 39(3):209–218, 2008.

[17] T. M. Chan and S. Har-Peled. Approximation algorithms for maximum independent set of pseudo-disks. *Discrete & Computational Geometry*, 48(2):373–392, 2012.

[18] G. J. Chang and G. L. Nemhauser. The k-domination and k-stability problems on sun-free chordal graphs. *SIAM Journal on Algebraic Discrete Methods*, 5(3):332–345, 1984.

[19] M. Chellali and T. W. Haynes. A note on the total domination number of a tree. *Journal of Combinatorial Mathematics and Combinatorial Computing*, 58:189–193, 2006.

[20] M. Chellali, T. W. Haynes, S. T. Hedetniemi, and T. M. Lewis. On ve-degrees and ev-degrees in graphs. *Discrete Mathematics*, 340(2):31–38, 2017.

[21] X. Chen, K. Yin, and T. Gao. A note on independent vertex–edge domination in graphs. *Discrete Optimization*, 25:1–5, 2017.

[22] S. Chitra and R. Sattanathan. Global vertex-edge domination sets in graph. In *International Mathematical Forum*, volume 7, pages 233–240, 2012.

[23] M. Chlebík and J. Chlebíková. Approximation hardness of dominating set problems in bounded degree graphs. *Information and Computation*, 206(11):1264–1275, 2008.

[24] V. Chvátal and C. McDiarmid. Small transversals in hypergraphs. *Combinatorica*, 12(1):19–26, 1992.

[25] B. N. Clark, C. J. Colbourn, and D. S. Johnson. Unit disk graphs. *Discrete mathematics*, 86(1-3):165–177, 1990.

[26] E. J. Cockayne, R. M. Dawes, and S. T. Hedetniemi. Total domination in graphs. *Networks*, 10(3):211–219, 1980.

[27] E. J. Cockayne, M. A. Henning, and C. M. Mynhardt. Vertices contained in all or in no minimum total dominating set of a tree. *Discrete mathematics*, 260(1-3):37–44, 2003.

[28] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to algorithms*. MIT press, 2009.

[29] G. K. Das, G. D. da Fonseca, and R. K. Jallu. Efficient independent set approximation in unit disk graphs. *Discrete Applied Mathematics*, 280:63–70, 2020.

[30] G. K. Das, M. De, S. Kolay, S. C. Nandy, and S. Sur-Kolay. Approximation algorithms for maximum independent set of a unit disk graph. *Information Processing Letters*, 115(3):439–446, 2015.

[31] G. K. Das, S. C. Ghosh, and S. C. Nandy. An efficient heuristic algorithm for 2d h-hops range assignment problem. In *IEEE Global Telecommunications Conference, 2004. GLOBECOM'04.*, volume 2, pages 1051–1055. IEEE, 2004.

[32] G. K. Das, S. C. Ghosh, and S. C. Nandy. Improved algorithm for minimum cost range assignment problem for linear radio networks. *International Journal of Foundations of Computer Science*, 18(03):619–635, 2007.

[33] M. De, G. K. Das, P. Carmi, and S. C. Nandy. Approximation algorithms for a variant of discrete piercing set problem for unit disks. *International Journal of Computational Geometry & Applications*, 23(06):461–477, 2013.

[34] E. DeLaViņa, Q. Liu, R. Pepper, B. Waller, and D. B. West. Some conjectures of graffiti. pc on total domination. 2007.

[35] I. Dinur and D. Steurer. Analytical approach to parallel repetition. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pages 624–633, 2014.

[36] M. Dorfling, W. Goddard, and M. A. Henning. Domination in planar graphs with small diameter ii. 2006.

[37] T. Erlebach, K. Jansen, and E. Seidel. Polynomial-time approximation schemes for geometric intersection graphs. *SIAM Journal on Computing*, 34(6):1302–1323, 2005.

[38] H. Eto, F. Guo, and E. Miyano. Distance-d independent set problems for bipartite and chordal graphs. *Journal of Combinatorial Optimization*, 27(1):88–99, 2014.

[39] H. Eto, T. Ito, Z. Liu, and E. Miyano. Approximability of the distance independent set problem on regular graphs and planar graphs. In *International Conference on Combinatorial Optimization and Applications*, volume 10043 of *LNCS*, pages 270–284. Springer, 2016.

[40] H. Eto, T. Ito, Z. Liu, and E. Miyano. Approximation algorithm for the distance-3 independent set problem on cubic graphs. In *International Workshop on Algorithms and Computation*, volume 10167 of *LNCS*, pages 228–240. Springer, 2017.

[41] G. D. da Fonseca, C. M. H. de Figueiredo, V. G. P. de Sá, and R. Machado. Linear time approximation for dominating sets and independent dominating sets in unit disk graphs. In *Approximation and Online Algorithms - 10th International Workshop, WAOA 2012*, pages 82–92, 2012.

[42] M. R. Garey and D. S. Johnson. The rectilinear steiner tree problem is np-complete. *SIAM Journal on Applied Mathematics*, 32(4):826–834, 1977.

[43] M. R. Garey and D. S. Johnson. *Computers and intractability: A guide to the theory of NP-completeness*. W. H. Freeman and company, 1978.

[44] M. R. Garey, D. S. Johnson, and L. Stockmeyer. Some simplified NP-complete problems. In *Proceedings of the sixth annual ACM symposium on Theory of computing*, pages 47–63. ACM, 1974.

[45] F. Gavril. Algorithms for minimum coloring, maximum clique, minimum covering by cliques, and maximum independent set of a chordal graph. *SIAM Journal on Computing*, 1(2):180–187, 1972.

[46] M. Gibson and I. A. Pirwani. Algorithms for dominating set in disk graphs: breaking the logn barrier. In *European Symposium on Algorithms*, pages 243–254. Springer, 2010.

[47] M. M. Halldórsson. Approximating discrete collections via local improvements. In *Proceedings of the sixth annual ACM-SIAM Symposium on Discrete Algorithms*, volume 95, pages 160–169, 1995.

[48] F. Harary and T. W. Haynes. Double domination in graphs. *Ars Combinatoria*, 55:201–214, 2000.

[49] M. Hasan, S. M. S. Hossain, Md. M. Rahman, and M. S. Rahman. Solving minimum hitting set problem and generalized exact cover problem with light based devices. *International Journal of Unconventional Computing*, 7(1-2):125–140, 2011.

[50] T. W. Haynes, S. Hedetniemi, and P. Slater. *Fundamentals of domination in graphs.* CRC press, 1998.

[51] T. W. Haynes, S. T. Hedetniemi, and P. J. Slater. Domination in graphs–advanced topics. *Marcel Decker Inc., New York*, 1998.

[52] T. W. Haynes and M. A. Henning. Trees with unique minimum total dominating sets. *Discussiones Mathematicae Graph Theory*, 22(2):233–246, 2002.

[53] M. A. Henning. A survey of selected recent results on total domination in graphs. *Discrete Mathematics*, 309(1):32–63, 2009.

[54] M. A. Henning and A. Yeo. *Total domination in graphs.* Springer, 2013.

[55] D. S. Hochbaum and W. Maass. Approximation schemes for covering and packing problems in image processing and VLSI. *Journal of the ACM (JACM)*, 32(1):130–136, 1985.

[56] J. Hopcroft and R. Tarjan. Efficient planarity testing. *Journal of the ACM (JACM)*, 21(4):549–568, 1974.

[57] B. Horoldagva, K. C. Das, and T. Selenge. On ve-degree and ev-degree of graphs. *Discrete Optimization*, 31:1–7, 2019.

[58] W. Hsu and G. L. Nemhauser. Algorithms for minimum covering by cliques and maximum clique in claw-free perfect graphs. *Discrete Mathematics*, 37(2-3):181–191, 1981.

[59] A. Itai, C. H.. Papadimitriou, and J. L. Szwarcfiter. Hamilton paths in grid graphs. *SIAM Journal on Computing*, 11(4):676–686, 1982.

[60] D. S. Johnson, M. R. Garey, and L. Stockmeyer. Some simplified np-complete graph problems. *Theoretical Computer Science*, 1(3):237–267, 1976.

[61] D. S. Johnson, M. Yannakakis, and C. H. Papadimitriou. On generating all maximal independent sets. *Information Processing Letters*, 27(3):119–123, 1988.

[62] B. Krishnakumari, M. Chellali, and Y. B. Venkatakrishnan. Double vertex-edge domination. *Discrete Mathematics, Algorithms and Applications*, 9(4):1–11, 2017.

[63] B. Krishnakumari and Y. B. Venkatakrishnan. Influence of the edge removal, edge addition and edge subdivision on the double vertex–edge domination number of a graph. *National Academy Science Letters*, 41(6):391–393, 2018.

[64] B. Krishnakumari and Y. B. Venkatakrishnan. The outer-connected vertex edge domination number of a tree. *Communications of the Korean Mathematical Society*, 33(1):361–369, 2018.

[65] B. Krishnakumari, Y. B. Venkatakrishnan, and M. Krzywkowski. Bounds on the vertex–edge domination number of a tree. *Comptes Rendus Mathematique*, 352(5):363–366, 2014.

[66] R. Laskar, J. Pfaff, S. M. Hedetniemi, and S. T. Hedetniemi. On the algorithmic complexity of total domination. *SIAM Journal on Algebraic Discrete Methods*, 5(3):420–425, 1984.

[67] J. Lewis. Vertex-edge and edge-vertex parameters in graphs. *Dissertation presented to Graduate School of Clemson University*, 2007.

[68] J. Lewis, S. T. Hedetniemi, T. W. Haynes, and G. H. Fricke. Vertex-edge domination. *Utilitas Mathematica*, 81:193–213, 2010.

[69] C. Liao and G. J. Chang. k-tuple domination in graphs. *Information Processing Letters*, 87(1):45–50, 2003.

[70] M. V. Marathe, H. Breu, H. B. Hunt, S. S. Ravi, and D. J. Rosenkrantz. Simple heuristics for unit disk graphs. *Networks*, 25(2):59–68, 1995.

[71] T. Matsui. Approximation algorithms for maximum independent set problems and fractional coloring problems on unit disk graphs. In *Japanese Conference on Discrete and Computational Geometry*, volume 1763 of *LNCS*, pages 194–200. Springer, 1998.

[72] G. J. Minty. On maximal independent sets of vertices in claw-free graphs. *Journal of Combinatorial Theory, Series B*, 28(3):284–304, 1980.

[73] K. Miura, S. Nakano, and T. Nishizeki. Grid drawings of 4-connected plane graphs. *Discrete & Computational Geometry*, 26(1):73–87, 2001.

[74] P. Montealegre and I. Todinca. On distance-d independent set and other problems in graphs with "few" minimal separators. In *International Workshop on Graph-*

Theoretic Concepts in Computer Science, volume 9941 of *LNCS*, pages 183–194. Springer, 2016.

[75] O. J. Murphy. Computing independent sets in graphs with large girth. *Discrete Applied Mathematics*, 35(2):167–170, 1992.

[76] T. Nieberg and J. Hurink. A PTAS for the minimum dominating set problem in unit disk graphs. In *International Workshop on Approximation and Online Algorithms*, volume 3879 of *LNCS*, pages 296–306. Springer, 2005.

[77] T. Nieberg, J. Hurink, and W. Kern. A robust PTAS for maximum weight independent sets in unit disk graphs. In *Graph-theoretic concepts in computer science*, volume 3353 of *LNCS*, pages 214–221. Springer, 2005.

[78] B. S. Panda and S. Paul. Liar's domination in graphs: Complexity and algorithm. *Discrete Applied Mathematics*, 161(7):1085–1092, 2013.

[79] B. S. Panda and S. Paul. A linear time algorithm for liar's domination problem in proper interval graphs. *Information Processing Letters*, 113(19):815–822, 2013.

[80] B. S. Panda, S. Paul, and D. Pradhan. Hardness results, approximation and exact algorithms for liar's domination problem in graphs. *Theoretical Computer Science*, 573:26–42, 2015.

[81] KWJ Peters. Theoretical and algorithmic results on domination and connectivity. *Dissertation presented to Graduate School of Clemson University*, 1987.

[82] J. Pfaff, R. Laskar, and S. T. Hedetniemi. Np-completeness of total and connected domination and irredundance for bipartite graphs. In *Tech. Rept. 428*. Clemson University Clemson, SC, 1983.

[83] S. Poljak. A note on stable sets and colorings of graphs. *Commentationes Mathematicae Universitatis Carolinae*, 15(2):307–309, 1974.

[84] M. S. Rahman, S. Nakano, and T. Nishizeki. Rectangular grid drawings of plane graphs. *Computational Geometry*, 10(3):203–220, 1998.

[85] Md. M. Rasheed, M. Hasan, and M. S. Rahman. Maximum neighbour voronoi games. In *International Workshop on Algorithms and Computation*, volume 5431 of *LNCS*, pages 93–104. Springer, 2009.

[86] D. Rautenbach and L. Volkmann. New bounds on the k-domination number and the k-tuple domination number. *Applied mathematics letters*, 20:98–102, 2007.

[87] R. Raz and S. Safra. A sub-constant error-probability low-degree test, and a sub-constant error-probability pcp characterization of np. In *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, volume 97, pages 475–484, 1997.

[88] J. M. Robson. Algorithms for maximum independent sets. *Journal of Algorithms*, 7(3):425–440, 1986.

[89] M. L. Roden and P. J. Slater. Liar's domination in graphs. *Discrete Mathematics*, 309(19):5884–5890, 2009.

[90] S. V. Siva Rama Raju and I. H. Nagaraja Rao. Complementary nil vertex edge dominating sets. *Proyecciones (Antofagasta)*, 34(1):1–13, 2015.

[91] P. J. Slater. Liar's domination. *Networks,*, 54(2):70–74, 2009.

[92] L. Sun. An upper bound for the total domination number. *J. Beijing Inst. Tech*, 4(2):111–114, 1995.

[93] R. E. Tarjan and A. E. Trojanowski. Finding a maximum independent set. *SIAM Journal on Computing*, 6(3):537–546, 1977.

[94] D. K. Thakkar and N. P. Jamvecha. About ve-domination in graphs. *Annals of Pure and Applied Mathematics*, 14(2):245–250, 2017.

[95] S. Thomassé and A. Yeo. Total domination of graphs and small transversals of hypergraphs. *Combinatorica*, 27(4):473–487, 2007.

[96] Z. Tuza. Covering all cliques of a graph. *Discrete Mathematics*, 86(1-3):117–126, 1990.

[97] E. J. v. Leeuwen. Optimization and approximation. 2009.

[98] L. G. Valiant. Universality considerations in VLSI circuits. *IEEE Transactions on Computers,*, 100(2):135–140, 1981.

[99] V. V. Vazirani. *Approximation algorithms.* Springer Science & Business Media, 2013.

[100] Y. B. Venkatakrishnan and H. N. Kumar. On the algorithmic complexity of double vertex-edge domination in graphs. In *International Workshop on Algorithms and Computation*, volume 11355 of *LNCS*, pages 188–198. Springer, 2019.

[101] B. Verweij and K. Aardal. An optimisation algorithm for maximum independent set with applications in map labelling. In *European Symposium on Algorithms*, volume 1643 of *LNCS*, pages 426–437, 1999.

[102] A. Vijayan and T. Nagarajan. Vertex-edge dominating sets and vertex-edge domination polynomials of wheels. *IOSR Journal of Mathematics*, 10(5):14–21, 2014.

[103] A. Vijayan and T. Nagarajan. Vertex-edge domination polynomial of graphs. *International Journal of Mathematical Archive*, 5(2):281–292, 2014.

[104] D. P. Williamson and D. B. Shmoys. *The design of approximation algorithms.* Cambridge university press, 2011.

[105] M. Xiao and H. Nagamochi. Exact algorithms for maximum independent set. *Information and Computation*, 255:126–146, 2017.

[106] P. Żyliński. Vertex-edge domination in graphs. *Aequationes mathematicae*, 93(4):735–742, 2019.

Publications from the Contents of the Thesis

Papers published/submitted in international journals:

[J1] Sangrm K. Jena, and Gautam K. Das , **Vertex-Edge Domination in unit disk graphs**, submitted to *Discrete Applied Mathematics (DAM), 2020 (minor correction needed as per review report).*

[J2] Sangram K. Jena, Ramesh K. Jallu, and Gautam K. Das, **On d-distance m-tuple (ℓ, r)-domination in graphs**, revised version submitted to *Information Processing Letters (IPL), 2019.*

[J3] Sangram K. Jena, Ramesh. K. Jallu, Gautam. K. Das, and Subash. C. Nandy, **Generalized Independent and Dominating Set Problems in Unit Disk Graphs**, submitted to *Discrete Applied Mathematics (DAM), 2020.*

Papers published/submitted in international conference proceedings:

[C1] Sangram K. Jena, Ramesh K. Jallu, Gautam K. Das, and Subash. C. Nandy, **The Maximum Distance-d Independent Set Problem on Unit Disk Graphs**, in *Frontiers in Algorithmics*, Lecture Notes in Computer Science, pp 68-80, 2018.

[C2] Sangram K. Jena, and Gautam K. Das, **Vertex-Edge Domination in Unit Disk Graphs**, in *Conference on Algorithms and Discrete Applied Mathematics (CALDAM)*, Lecture Notes in Computer Science, pp. 67-78, 2020.

[C3] Sangram K. Jena, and Gautam K. Das, **Total Domination in Unit Disk Graphs**, https://arxiv.org/pdf/2007.11997.pdf *(draft version).*