GENERALIZED DOMINATION IN GRAPHS WITH APPLICATIONS IN WIRELESS NETWORKS

A Dissertation

by

JE SANG SUNG

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

DOCTOR OF PHILOSOPHY

Chair of Committee,	Sergiy Butenko
Committee Members,	Amarnath Banerjee
	Kiavash Kianfar
	Natarajan Sivakumar
Head of Department,	Cesar O. Malave

December 2013

Major Subject: Industrial Engineering

ABSTRACT

The objective of this research is to study practical generalization of domination in graphs and explore the theoretical and computational aspects of models arising in the design of wireless networks. For the construction of a virtual backbone of a wireless ad-hoc network, two different models are proposed concerning reliability and robustness. This dissertation also considers wireless sensor placement problems with various additional constraints that reflect different real-life scenarios.

In wireless ad-hoc network, a connected dominating set (CDS) can be used to serve as a virtual backbone, facilitating communication among the members in the network. Most literature focuses on creating the smallest virtual backbone without considering the distance that a message has to travel from a source until it reaches its desired destination. However, recent research shows that the chance of loss of a message in transmission increases as the distance that the message has to travel increases. We propose CDS with bounded diameter, called dominating s-club (DsC) for $s \geq 1$, to model a reliable virtual backbone.

An ideal virtual backbone should retain its structure after the failure of a certain number of vertices. The issue of robustness under vertex failure is considered by studying k-connected m-dominating set. We describe several structural properties that hold for $m \geq k$, but fail when $m < k$. Three different formulations based on vertex-cut inequalities are shown depending on the value of k and m. The computational results show that the proposed lazy-constraint approach compares favorably with existing methods for the minimum connected dominating set problem (for $k = m = 1$). The experimental results for $k = m = 2, 3, 4$ are presented as well.

In the wireless sensor placement problem, the objective is often to place a minimum number of sensors while monitoring all sites of interest, and this can be modeled by dominating set. In some practical situations, however, there could be a location where a sensor cannot be placed because of environmental restrictions. Motivated by these practical scenarios, we introduce varieties of dominating set and the corresponding optimization problems. For these new problems, we consider the computational complexity, mathematical programming formulation, and analytical bounds on the size of structures of interest. We solve these problems using a general commercial solver and compare its performance with that of simulated annealing.

To my parents

ACKNOWLEDGEMENTS

First of all, I would like to thank Sergiy Butenko, who has been a great academic adviser and mentor while I was pursuing M.S. and Ph.D. degrees at Texas A&M University. Without his inspirations for research, patience and financial support, I would not have been able to finish this long journey. I also would like to thank my committee members Amarnath Banerjee, Kiavash Kianfar, and Natarajan Sivakumar.

Most of the work presented in this dissertation was accomplished while I was working as an operations research summer intern at University of Florida Research and Engineering Education Facility and Air Force Research Lab. I would like to thank the summer internship program directors Eduardo L Pasiliao Jr and Vladimir Boginski for giving me such great opportunities to work in an environment perfectly suited for research. I also would like to thank the other summer interns Jose Walteros, Chrysafis Vogiatzis, Matthew T Faatz, Behdad Beheshti, Gabriel Zenarosa, and Yiming Wang who made the summers in Florida productive and enjoyable. I would especially like to thank Austin Buchanan for being a great roommate and a collaborator during the internship period. It was a pleasurable experience to discuss ideas, prove/disprove theorems, and tackle challenging problems together.

I would like to thank my family, Youn Ok Sung, Kyung Ok Sung, Hak Jun Kim, Jung Gu Park, Kyung Gen Park, Kyung Min Park, and Ji Yul Kim. Their support for me and dedication to my parents during my absence was the key ingredient which enabled me to finish my degrees. Also, I would like to thank my in-laws In Chol Suh, Bo Sook Suh, Yoon Hee Suh, and Harrison Steadman for making me part of this wonderful family.

Lastly, my utmost appreciation goes to my wife, Jean Hee Suh, who was right next to me while I was passing through the seemingly endless dark tunnel. Her love, unyielding positiveness, understanding and support in every aspect of my day-to-day life were the core elements of this work.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

The objective of this research is to study practical generalization of domination in graphs and explore the theoretical and computational aspects of proposed models with applications in design of reliable and robust wireless network topology.

In a wired network, there is a *network backbone* that interconnects various components of a network, providing a channel for the exchange of information. However, in a wireless network, there is no such fixed structure but each node (or vertex) has its own wireless communication ability. A wireless ad-hoc network can be modeled using a graph consisting of nodes and edges. A node represents a wireless communication device and an edge linking two nodes indicates that they can communicate directly. Figure 1.1 is an illustration of a such graph with 17 nodes.

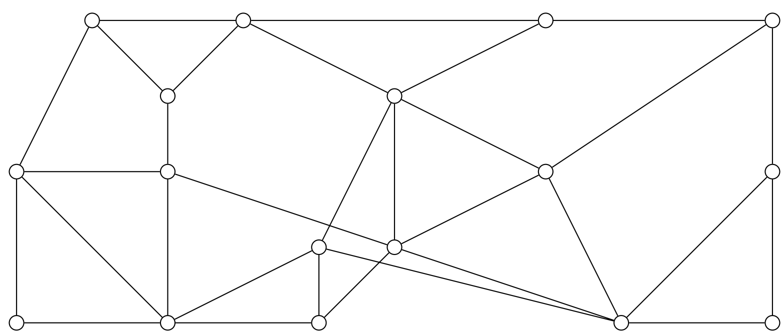


Figure 1.1: A graph modeling a wireless network.

One of the major disadvantages of a wireless network compared to a wired network is that each vertex has only limited transmission range and limited power. Two nodes in a network must be within the transmission range of each other for them to directly communicate, and greater transmission range requires greater power consumption.

When direct communication is infeasible for two nodes, a message must be relayed through intermediate nodes between them. The collection of intermediate nodes that are in charge of relaying messages is called a *virtual backbone* of the wireless network. The construction of a virtual backbone has drawn significant attention in the computer science and operations research communities and there exist numerous articles introducing methods of virtual backbone creation. The set of dark vertices in Figure 1.2 is a virtual backbone of the network. Note that the induced subgraph of these vertices is connected and every white node is adjacent to at least one dark node. In a graph with nodes and edges, a *dominating set* is a set of nodes such that each node in the graph is either in the dominating set or has a neighboring node in the dominating set. If a dominating set induces a connected graph, then it is a *connected dominating set*. The set of dark nodes in Figure 1.2 is a connected dominating set in the graph. For formal definitions of varieties of dominating set, refer to Section 2.2.

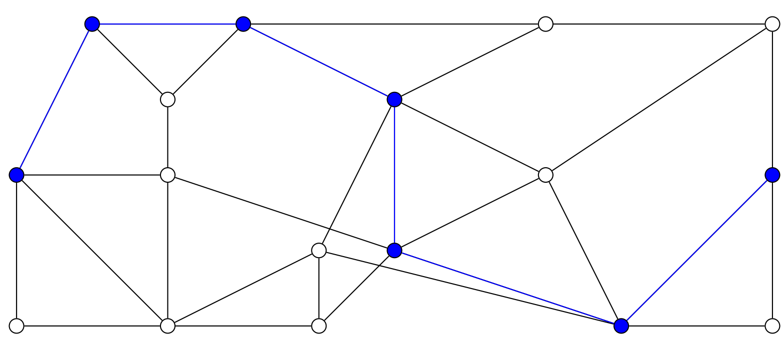


Figure 1.2: A virtual backbone of a wireless network with diameter 6.

In terms of virtual backbone construction for wireless network, there are a few desirable properties that need to be considered, such as the size of the virtual backbone (small number of nodes in a virtual backbone is preferred), latency (short transmission time is desired), and robustness (virtual backbone should remain functioning after node failures). In early stages of study, researchers mainly focused on the construction of virtual backbones with small sizes because of the following reasons. First, a small virtual backbone reduces the interference between nodes. Second, the smaller the virtual backbone, the easier it is to maintain. Third, a small virtual backbone tends to provide more efficient routing and requires less control messages. We can find the smallest virtual backbone by solving the minimum connected dominating set problem. However, connected dominating set ensured only the bare-minimum functional structure. Mohammed et al. [80] shows that the chance of loss of a message in transmission increases as the distance that message has to travel increases. This implies that a virtual backbone of small diameter is more desirable if all other conditions are the same. Moreover, low latency can be ensured by constructing a virtual backbone with a small *diameter*, where the diameter of a graph is defined to be the length of the longest shortest distance (hop distance) between every pair of nodes in the graph. In Figure 1.2, for instance, the diameter of the virtual backbone is 6. We can construct a virtual backbone of the same size as in Figure 1.3, but with a smaller diameter equal to 4.

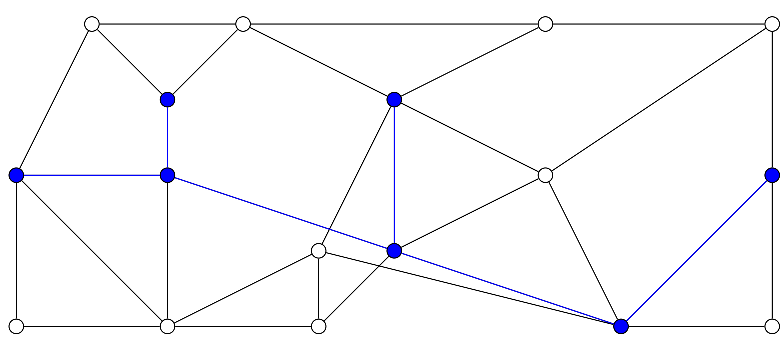


Figure 1.3: A virtual backbone of diameter 4.

As mentioned above, reducing the diameter of a virtual backbone increases the reliability of data transmission between nodes in the network. From that perspective, it would be the best if every member of the virtual backbone is adjacent to every other member of the virtual backbone. Cozzens and Kelleher [27] considered a connected dominating set of diameter 1, i.e., dominating set which induces a complete graph, and called it a *dominating clique*. Dominating cliques have numerous applications, but the original concept stems from the context of social network analysis [67]. For a graph which models a social network, a node represents an actor and an edge between two nodes indicates that the two actors corresponding to the incident nodes know each other. On the other hand, for a graph modeling a wireless network, a dominating clique represents a core group of nodes in the network that not only serves as a virtual backbone, but also has the ability for every member of the core group to directly communicate with every other member of the core group. However, clique is overly restrictive, moreover, not every graph has a dominating clique. In Section 3, we propose a more relaxed but still reliable virtual backbone structure called *dominating s-club* by combining two graph theoretic concepts: dominating set and s-club. Note that s-club is a form of clique relaxation, and more details about clique relaxations are presented in Section 2.3. We formally define dominating s-club and minimum dominating s-club problem in Section 3. Brandstadt and Kratsch [17] shows that it is NP-complete [43] to check whether a graph has a dominating clique or not by using reduction from MONOTONE 3SAT. We generalize this result and demonstrate that checking the existence of a dominating s-club in a graph is NP-complete for any fixed positive integer s using reduction from 3SAT. Kelleher and Cozzens [67] show that there exists a dominating clique in connected graphs that are both C_5-free and P_5-free, where C_5 and P_5 are a cycle and a path of 5 vertices, respectively. We show that this can be generalized to dominating s-club and prove that there exists a dominating s-club if there is no induced C_{s+4} and no induced P_{s+4} for any fixed positive integer s.

Another desirable characteristic of a virtual backbone is robustness under node failure. As mentioned above, connected dominating set ensures the bare-minimum requirement to serve as a virtual backbone. In Figure 1.2 or 1.3, it is easy to see that if a dark node in the connected dominating set fails to work, it cannot serve as a virtual backbone of the network anymore. A straightforward remedy for this issue is to include redundant nodes to the virtual backbone. In terms of adding redundancy, there are two conditions that must be considered. One is connectivity and the other is dominance. After failure of a certain number of nodes, we want the virtual backbone to remain connected, and the nodes outside of virtual backbone to remain dominated. A *k-connected m-dominating set* $(k, m \geq 1)$, which generalizes the connected dominating set, perfectly satisfies this requirement. A graph is k-connected if it remains connected after the failure of $k - 1$ nodes. A node is m-dominated if it has m neighboring nodes in the dominating set. As its name implies, a virtual backbone which is modeled by a k-connected m-dominating set remains connected after the failure of up to $k - 1$ nodes.

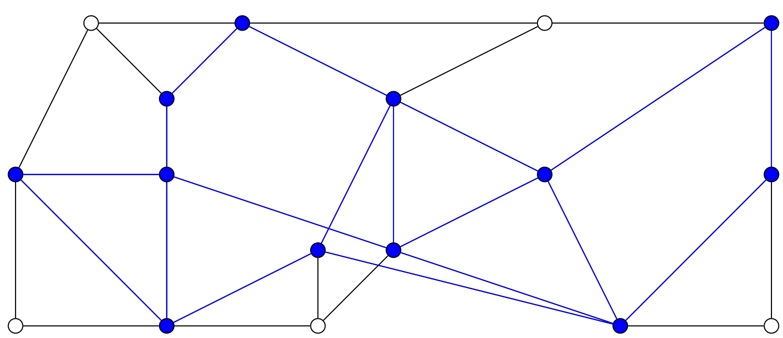


Figure 1.4: A 2-connected 2-dominating set.

In Figure 1.4, the set of dark vertices forms a 2-connected 2-dominating set. The robustness of a k-connected m-dominating set depends on the values of k and m and it is not hard to see that the existence of a k-connected m-dominating set in a graph depends on the values k and m. For instance, if there is an articulation vertex in a graph, then there is no 2-connected m-dominating set for any positive integer m. For a connected graph, a vertex is an articulation vertex if its removal renders the graph disconnected. So far, minimum k-connected m-dominating set has been studied only by computer scientists and several heuristic algorithms for general graphs and approximation algorithms for unit disk graphs have been introduced. However, to the best of our knowledge, no exact approach has been published for this problem. In Section 4, an integer programming approach for this problem is presented. Note that the robustness is a desirable property but it is not free. In fact, as shown in Section 4, ensuring robustness requires a strictly more-costly solution. Three different mathematical programming formulations based on vertex-cut inequalities are proposed. A vertex-cut, which is also refereed to as a separating set, is a set of vertices in a graph whose removal renders the graph disconnected. Note that an articulation vertex is a vertex-cut of size one. The separation problem for vertex-cut inequalities is weighted vertex-connectivity problem and it can be solved in polynomial time. We can generalize vertex-cut and vertex-cut inequalities by introducing r-robust vertex-cut and r-robust vertex-cut inequalities. For numerical experiment, lazy-constraint approach is used and the test results show that the proposed method compares favorably with existing approaches for minimum 1-connected 1-dominating set (or minimum connected dominating set) problem. Computational results comparing minimum k-total dominating set problem and minimum k-connected k-dominating set problem for $k = 2, 3, 4$ also reveal that the connectivity requirement is not burdensome.

Resource allocation problems can be modeled using graphs with vertices and edges. A vertex represents a location on which a facility can be installed and there is an edge between two vertices i, j if resource in location i can serve location j, and vice versa. In many cases, we want to install a minimal number of resources while satisfying the needs of all sites. For instance, consider a wireless sensor placement problem, where each vertex of the graph represents a site that must be monitored. Two vertices are connected by an edge if both of the corresponding sites can be monitored by one sensor installed in one of these two sites. In order to monitor all sites of interest while minimizing the number of sensors needed, we can solve the minimum dominating set problem on the graph. The information collected from these sensors could be used to initiate some other actions such as cuing video cameras, unmanned aerial vehicles and calling engineers. In Section 5, several varieties of the classical dominating set are introduced considering some practical scenarios. For instance, there could be a location where a sensor cannot be placed because of environmental restrictions such as frequent (or on-going) chemical reaction, extreme humidity, or temperature. On the other hand, there could be a site which is more important than others such that if there is no sensor on it, more than one sensor in the neighboring locations is needed. This redundancy provides the system robustness under node failure. The difference between the dominating set and the considered variations of dominating set is the domination requirement assigned to each vertex. In the classical minimum dominating set problem, all vertices have domination requirement 1, implying that if a vertex is not in the dominating set, it should be dominated by one of its neighboring vertices in the dominating set. However, in the variations of the dominating set proposed in Section 5, domination requirement assigned to each vertex is not uniform. For every variation of dominating set introduced in Section 5, we assume that the domination requirement of each vertex in the considered graph is determined a priori depending on the importance of the sites that each vertex represents. Complexity of the decision version of the corresponding optimization problems is analyzed showing that these variations are hard on their own respect. We consider integer programming formulation of each problem and also establish some basic properties of the corresponding polyhedra. We also develop several analytical bounds on the size of structures of interest. Results of numerical experiments using CPLEX 12.1® are also reported. We compare the performance of simulated annealing and CPLEX 12.1® on random unit disk graphs and also on some standard test instances.

2. BACKGROUND

This section provides some background that is required to understand the following sections. Section 2.1 introduces basic notations and definitions used in this dissertation. Section 2.2 formally defines the classical dominating set and some of its variations. A brief introduction of clique and clique relaxations are presented in section 2.3.

2.1 Graph theory

In this section, we briefly introduce the notations and definitions used in this dissertation. For a comprehensive introduction to graph theory, readers are refereed to [10] or [101]. Throughout this dissertation, we consider a finite, simple, undirected graph which is denoted by $G = (V, E)$ where $V = \{1, 2, \ldots, n\}$ and $(i, j) \in E$ when vertices i and j are joined by an edge with $|E| = m$. *Order* and *size* of a graph is the number of vertices and the number of edges in the graph, respectively. We use $V(G)$ and $E(G)$ to represent the vertex set and the edge set of a graph G, respectively. Whenever the graph under consideration is clear from the context, we suppress G and use V and E to represent the vertex set and the edge set. A graph G is called a *null* graph if $V = E = \emptyset$ and a *trivial* graph if $E = \emptyset$. K_n denotes a *complete graph* on n vertices in which every pair of distinct vertices is connected by a unique edge. A *bipartite graph (or bigraph)* is a graph such that the vertices can be divided into two disjoint subsets U_1 and U_2 and every edge connects a vertex in U_1 to one in U_2. A *complete bipartite graph* with bipartitions of size p and q is denoted by $K_{p,q}$. Note that the graph $K_{1,n}$ is called a *star graph.* A_G (or A) is used to represent the *adjacency matrix* of G. A_G is a symmetric $0, 1$-matrix of order $n \times n$ with $a_{i,j} = 1$ if and only if $(i, j) \in E$. For a vertex $i \in V$, the (open) neighbor of i is

$N(i) = \{j \in V : (i,j) \in E\}$, and the closed neighborhood of i is $N[i] = N(i) \cup \{i\}$. Clearly, the set of *non-neighbors* of a vertex $i \in V$ is given by $V \setminus N[i]$. The degree of $i \in V$ is denoted by $deg_G(i)$ and is given by $deg_G(i) = |N(i)|$. For a subset $S \subset V$, $G[S]$ indicates the subgraph of G induced by S, i.e. $G[S] = (S, E \cap S \times S)$. Degree of a vertex $i \in S \subset V$ in $G[S]$ is denoted by $deg_{G[S]}(i)$ and $deg_{G[S]}(i) = |N(i) \cap S|$. We use $\Delta(G)$ and $\delta(G)$ to represent the maximum and minimum degree of a vertex in G, respectively. The distance between two vertices $i, j \in V$ in G, denoted by $d_G(i,j)$, is the length of a shortest path between i and j in G (measured in the number of edges). For $i, j \in S \subset V$, the distance between i and j in the induced subgraph of S is denoted by $d_{G[S]}(i,j)$. By convention, if there is no path between i and j, then the distance between the two vertices is infinite. The diameter of a graph G is denoted by $diam(G) = max_{i,j \in V} d_G(i,j)$. A sequence of $k+1$ distinct vertices $u_0, u_1, u_2, \ldots, u_k$ together with edges $(u_0, u_1), (u_1, u_2), \ldots, (u_{k-1}, u_k)$ is called a path of $k+1$ vertices and is denoted as P_{k+1}. If $u_0 = u_k$ but $u_1, u_2, \ldots, u_k$ are distinct, then it is called a cycle of k vertices and is denoted as C_k.

Definition 1. *(Clique) Given a graph G, a set $S \subset V$ is a clique if $G[S]$ is complete.*

Definition 2. *(Independent set) Given a graph G, a set $S \subset V$ is an independent set if there are no edges in $G[S]$.*

The *maximum clique problem* for a graph G is to find the largest clique of the graph. The size of the largest clique is *clique number* and is denoted by $w(G)$. The maximum independent set problem is to find an independent set of maximum cardinality. The size of the largest independent set of a graph G is called the *independence number* and is denoted by $\alpha(G)$. Note that $S \subset V$ is a clique in G if and only if S is an independent set in the complement graph of G.

A subset $C \subset V$ of a connected graph G is called a *vertex-cut* if $G[V \setminus C]$ is disconnected. For a graph G which is not complete, the *vertex connectivity* (or simply *connectivity*) of G is the size of minimal vertex-cut and is denoted by $\kappa(G)$. For a positive integer k, a graph G is *k-vertex-connected* (or simply *k-connected*) if its vertex connectivity is at least k. This implies that a graph G is k-connected if there is no vertex-cut of size $k-1$. By convention, the connectivity of a complete graph with n vertices is $n-1$, i.e., $\kappa(K_n) = n-1$. Alternatively, we can say that a connected graph G is k-vertex-connected if there exist k vertex-disjoint paths between every pair of distinct vertices. This definition also applies to a complete graph. Menger's theorem [79] is a basic result of connectivity in finite undirected graph. The vertex-connectivity version of Menger's theorem for a finite undirected graph $G = (V, E)$ is the following. For two vertices $x, y \in V$ that are not adjacent, the size of the minimum vertex-cut for x and y is equal to the maximum number of pairwise vertex independent paths from x to y.

2.2 Dominating set

A set $S \subseteq V$ is said to be *dominating* if each vertex $v \in V \setminus S$ has a neighbor in S. Minimum dominating set (MDS) problem is to find a dominating set of smallest cardinality. The size of the smallest dominating set is called the *domination number* $\gamma(G)$. An interesting application of this problem is in social network analysis. One might be interested in finding a set of key players in a social network for the purpose of optimal diffusion of information [12, 67, 66, 68, 99]. Another popular application of dominating set is in wireless sensor placement problem. Each vertex of the graph $G = (V, E)$ represents a site that must be monitored and two vertices are connected by an edge when both of the corresponding sites can be monitored by a single sensor installed in one of the two sites. The goal is to install the minimum number of sensors while monitoring all sites of interest. This can be achieved by solving the minimum dominating set problem. MDS problem is one of the most fundamental combinatorial optimization problems and two books from Haynes et al. [52, 51] provide a comprehensive overview of domination in graphs. These two books cover almost all subjects concerning domination in graphs published before 1997 such as variations of domination, analytical bounds, computational complexity, and algorithms. It has been shown that the MDS problem is NP-hard in general graphs [43] and it remains NP-hard even when restricted to bipartite graphs [45]. Booth and Johnson [11] show that the problem is also NP-hard in chordal graphs. A graph $G = (V, E)$ is chordal if each of its cycles of four or more nodes has a chord, that is an edge linking two nodes that are not adjacent in the cycle.

Bouchakour and Mahjoub [14] consider properties of dominating set polytope for the first time and introduces a decomposition technique for graphs which are decomposable by one-node cutsets. Bouchakour et al. [13] studies the dominating set polytope in the class of graphs that decompose by vertex-cut of size one, where the pieces are cycles. It shows that one can obtain a complete description of the polytope by a system of inequalities when the graph is a cycle. A variety of approaches have been applied to solve MDS problem. Papers [38, 47] develop exact algorithms for the MDS problem in general graphs and [44] presents exact algorithms to solve MDS problem on some graph classes. It is not difficult to understand that if a graph $G = (V, E)$ is disconnected, we can tackle each connected component separately. Various preprocessing techniques are introduced in [2] to reduce the size of the input graph and the efficiency of those techniques are empirically shown in [60]. A linear algorithm for the domination number of a tree is demonstrated in [25]. Farber [35] presents polynomial time algorithms to find minimum weighted dominating sets and independent dominating sets in strongly chordal graphs. A chordal graph $G = (V, E)$ is strongly chordal if every cycle of even length (≥ 6) in G has an odd chord.

In optimization, the term "*heuristic*" refers to a technique for finding a good sub-optimal solution. Even though fast technical advancement in recent years brought us significant computational power, most of the optimization problems are still hard to solve optimally. Usually, a heuristic finds an initial solution and iteratively moves to a better solution using local search procedures if such solution exists. The searching procedure terminates when the first local optimum is reached. Note that heuristics do not guarantee finding a global optimum. In an effort to overcome the drawbacks of heuristics, *metaheuristics* typically combine some heuristics with a method of orientation towards solutions which are usually better than the ones provided by local search. For large and hard optimization problems, well-chosen heuristics and metaheuristics can provide high quality solutions within a reasonable solution time. For a comprehensive survey about metaheuristic algorithms, refer to [89, 46].

Sanchis [90] introduces several heuristic algorithms on MDS problem and provides experimental results on random graphs and also on graphs with known domination number. In order to generate instances with known domination number, techniques presented in [91, 92] are used. Papers [56, 63] presents an enhanced ant colony meta-heuristic based on "tournament" for MDS problem and presents the performance of the proposed method on some benchmark instances. Hybrid genetic algorithm [53] and simulated annealing [54] with stochastic local search are applied to tackle MDS problem. Note that simulated annealing has been successfully applied on numerous optimization problems [61, 62, 70].

Varieties of domination. Given a minimum dominating set $S \subset V$ of a graph $G = (V, E)$, [37] shows that one can always remove two edges from the graph so that S is no longer a dominating set for the graph. In order to obtain greater degree of assurance, we need a greater degree of domination. A vertex $v \in V \setminus S$ is *k-dominated* if it is dominated by at least k vertices in S, that is, $|N(v) \cap S| \geq k$. If all vertices in $V \setminus S$ are k-dominated by S, then S is called a k-dominating set. The size of the smallest k-dominating set is called the k-domination number and is denoted by $\gamma_k(G)$. Note that the 1-domination number $\gamma_1(G)$ is the usual domination number $\gamma(G)$. Harary and Haynes [50] studies the case for $k = 2$. Extending analytical results for domination number, [20, 88] provide several analytical bounds for k-domination number. Corneil and Perl [26] investigate the complexity of k-Dominating Set on various subclasses of perfect graphs such as comparability graphs, chordal graphs, bipartite graphs, split graphs, cographs, and k-trees.

Definition 3. *(Total-dominating set) Given a graph $G = (V, E)$, a set $S \subseteq V$ is said to be* total dominating *if each vertex $v \in V$ has a neighbor in S.*

The size of the smallest total dominating set is called the *total domination number* and is denoted by $\gamma^t(G)$. Note the difference between dominating set and total dominating set. In total dominating set, every vertex must be dominated by one of its neighboring vertices.

Definition 4. *(k-Total-dominating set) Given a graph $G = (V, E)$ and a positive integer k, a set $S \subset V$ is a k-total-dominating set if every vertex in V is adjacent to at least k vertices in S.*

The k-total-domination number $\gamma^t_k(G)$ is the size of the smallest k-total-dominating set of G. Note that in Section 4, we initially solve the minimum k-total-dominating set problem and add connectivity cuts as needed to solve the minimum k-connected k-dominating set problem.

Definition 5. *(Connected dominating set) Given a graph $G = (V, E)$, a dominating set $S \subset V$ which induces a connected graph is a* connected dominating set *(CDS).*

It is straightforward to see that only connected graphs have a CDS. The minimum cardinality of a CDS is called the connected domination number $\gamma_c(G)$. Clearly, $\gamma(G) \leq \gamma_c(G)$. For a graph G, if there is a universal vertex which is adjacent to every other vertex in G, then $\gamma(G) = 1$ and, obviously, $\gamma(G) = \gamma_c(G)$. The minimum connected dominating set (MCDS) problem and the closely related maximum leaf spanning tree problem have been receiving significant attention in literature. Clark et al. [24] shows that many standard graph theoretic problems remain NP-complete on unit disk graphs, including DOMINATING SET and CONNECTED DOMINATING SET, where DOMINATING SET and CONNECTED DOMINATING SET represent the decision version of MDS and MCDS problem, respectively. In order to solve MCDS problem, a wide variety of methods has been applied such as exact approaches [34, 78, 94, 40, 41, 39], approximation algorithms [48, 76], and polynomial-time approximation schemes for unit disk graphs [58, 23] and unit ball graphs [107]. For a comprehensive review of CDS construction methods, refer to [32].

Definition 6. *(Connected k-total-dominating set) Given a graph $G = (V, E)$ and a positive integer k, a set $S \subset V$ is a connected k-total-dominating set if S is a k-total-dominating set and $G[S]$ is connected.*

Connected k-total-domination number $\gamma_k^{c,t}(G)$ is the size of the smallest connected k-total-dominating set of G.

Creation of virtual backbone. Many routing protocols for wireless ad-hoc networks based on CDS have been proposed in literature and most of them introduce heuristics or approximation algorithms [4, 7, 22, 28, 30, 57, 86, 95, 102, 103, 69, 100, 104, 97, 106, 96]. A virtual backbone modeled by CDS could be vulnerable to data loss in transmission if the diameter of CDS is long [80]. Recently, CDS with bounded diameter is proposed as a reliable virtual backbone and some heuristics are known [69].

CDS is also inherently vulnerable under vertex failure and recent articles have considered the construction of fault-tolerant virtual backbone structure by considering k-connected m-dominating set [29, 97, 104, 100]. Given a graph G and two positive integers k and m, a set $S \subset V$ is a k-connected m-dominating set if the following two conditions are satisfied. (i) The subgraph induced by S (i.e. $G[S]$) is k-connected, (ii) each vertex not in S has at least m neighbors in S. Dai and Wu [29] proposes a k-connected k-dominating set as a robust virtual backbone structure and introduces several heuristics. Thai [97] studies a general fault-tolerant CDS and proposes approximation algorithms for 1-connected m-dominating set and k-connected k-dominating set in heterogeneous networks. Wu and Li [104] proposes a distributed algorithm with low message complexity to construct a k-connected m-dominating set and shows that the proposed algorithm has a constant factor approximation ratio when the maximum vertex degree is a constant. Wang et al. [100] proposes a constant factor approximation algorithm of 2-connected virtual backbone and [42] develops a distributed algorithm for 2-connected m-dominating set.

2.3 Clique relaxation

As defined in Section 2.1, a clique is a subset of vertices inducing a subgraph which is complete. Thus, clique provides a very robust structure in the sense that all pairs of vertices are adjacent to each other and the deletion of any vertex does not destroy this structure. The decision version of the maximum clique problem can be stated as follows.

CLIQUE: Given a graph $G = (V, E)$ and a positive integer k, does there exist a clique of size $\geq k$ in G?

CLIQUE is shown to be NP-complete [43] and extensive research has been performed introducing algorithms and analytical bounds. For a comprehensive review of the maximum clique problem, refer to [9]. In many practical applications, however, clique is overly restrictive because it may not be possible to find a network with all possible connections. Thus, several clique relaxations have been introduced by relaxing certain properties of clique while possessing some clique-like properties. There are three major types of relaxations: distance based, degree based, and density based relaxations. Refer to [84] for a comprehensive reference on clique relaxation models.

s-Clique and s-club. In distance based relaxations, we relax the condition of the distance between each pair of vertices to be 1. *s-Clique* [77] and *s-club* [81] belong to this category. An s-clique is a subset of vertices with pairwise distance at most s in the graph and s-club is a subset of vertices inducing a subgraph of diameter at most s. For a fixed positive integer s, the s-CLIQUE (s-CLUB) problem is defined as follows.

s-CLIQUE (s-CLUB): Given a graph $G = (V, E)$ and a positive integer k, does there exist an s-clique (s-club) of size $\geq k$ in G?

Both s-CLIQUE and s-CLUB are known to be NP-complete [6, 16]. Integer programming approaches for maximum s-club problem with $O(n^{s+1})$ number of entities are proposed in [6, 16]. As an alternative approach, [98] proposes a new linear binary formulation with $O(sn^2)$ number of entities by using the special properties of an s-club. Bourjolly et al. [15] describes some properties of s-club and propose three heuristics for maximum s-club problem. Carvalho et al. [21] present valid inequalities for the 2-club polytope and some conditions for these inequalities to be facets. Using these inequalities, they introduce a strengthened formulation of maximum 2-club problem and a cutting plane algorithm. Almeida and Carvalho [3] shows two formulations of maximum 3-club problem: one is compact and the other has non-polynomial number of constraints. They obtain new upper bound on the size of maximum 3-club improving the optimum 3-clique number bound. New families of valid inequalities of 3-club polytope are introduced and used for computational experiments to strengthen the LP relaxations.

s-Plex. In a degree based relaxation, we relax the condition that the degree of all vertices in a clique S of size $|S| = z$ to be $z - 1$ and *s-plex* [93] belong to this category. In other word, a set $S \subset V$ is an s-plex if the degree of each vertex in S from the induced subgraph $G[S]$ is at least $|S| - s$. Balasundaram et al. [5] show that s-Plex, the decision version of the maximum s-plex problem, is NP-complete for any constant positive integer s.

γ-Clique. In a density based relaxation, we relax the requirement that the edge density of the subgraph induced by a clique to be 1 and γ-clique [1] is in this category. Pattillo et al. [83] show that for any fixed real $\gamma \in (0, 1)$, the γ-Clique is NP-complete. Note that for $s = \gamma = 1$, an s-clique, s-club, and γ-dense subgraphs are equivalent to a clique.

Dominating clique. Given a graph $G = (V, E)$, a *dominating clique* is a set $S \subset V$ such that S is a dominating set and the induced subgraph $G[S]$ is complete. Minimum dominating clique problem seeks to find the smallest dominating clique and the size of the smallest dominating clique is called dominating clique number $\gamma_{cl}(G)$. This problem is first introduced by Cozzens et al. [27] and the decision version of this problem is shown to be NP-complete. Kratsch et al. [73] show that a chordal graph has a dominating clique *if and only if* it has diameter at most 3. Kratsch and Liedloff [74] introduce an $O(1.3387^n)$ time and polynomial space algorithm that either computes a minimum dominating clique or finds that the graph has no dominating clique for an input graph on n vertices. We can think of the vertices in the dominating clique to be a communication core of the network. A dominating clique provides an extremely robust and reliable virtual backbone structure in wireless communication network in the sense that the core members can communicate directly and a message from any source to any destination can be transmitted within 3-hop. A dominating clique, however, can be overly restrictive since not all graphs have a dominating clique.

3. CONNECTED DOMINATING SET WITH BOUNDED DIAMETER

In a wireless ad-hoc network, a virtual backbone facilitates the communication among the members in the network. Most articles in literature concerning the creation of a virtual backbone focus on constructing a small virtual backbone without considering the distance that a message has to travel until it reaches its desired destination. However, recent research shows that the chance of loss of data in wireless transmission increases as the transmission distance increases. Thus, constructing a virtual backbone with small diameter is critical. In terms of reachability, it would be best if every member of the virtual backbone is adjacent to every other member of the virtual backbone. Dominating clique perfectly satisfies this requirement. In Figure 3.1, the set of dark vertices forms a dominating clique for the graph. Dominating clique represents a core group of devices in the network that not only serves as a virtual backbone, but also has the ability for every member of the core group to directly communicate with every other member of the core group. However, unfortunately, clique is overly restrictive, and not every graph has a dominating clique. In this chapter, we consider a more relaxed but still reliable virtual backbone structure called dominating s-club. Brandstadt and Kratsch [17] shows that it is NP-complete to check whether a graph has a dominating clique. In Section 3.2, we generalize this result and demonstrate that checking the existence of a dominating s-club in a graph is NP-complete for any fixed positive integer s. Moreover, we show that it is NP-hard to solve minimum dominating s-club problem, even if restricted to graphs which are known to have a dominating clique. Kelleher and Cozzens [67] show that there ex-

[*]Parts of this section are based on the research described in Austin Buchanan, Je Sang Sung, Vladimir Boginski, Sergiy Butenko: On connected dominating sets of restricted diameter. Submitted to European Journal of Operational Research

ists a dominating clique in connected graphs that are both C_5-free and P_5-free. We show that this can be generalized to dominating s-club and prove that there exists a dominating s-club if there is no induced C_{s+4} and no induced P_{s+4} for any fixed positive integer s. In Section 3.5, a compact mathematical programming formulation is used to solve the problem. Valid inequalities and variable fixing techniques are introduced. In terms of the size of the smallest dominating s-club, a comparison with some variations of dominating set is presented. Computational experiment results in section 3.7 demonstrate that it is generally most beneficial to apply all of the proposed valid inequalities and variable fixing techniques.

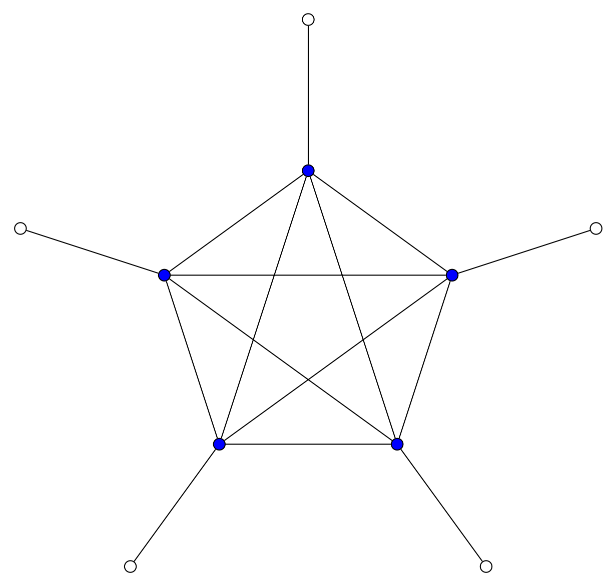


Figure 3.1: A dominating clique in a graph.

3.1 Related work

In wired network, every member of a network is physically connected through wires and each member of a network can communicate through this connection, even through they are located far apart. This physical connection is typically called a *backbone*. On the contrary, in wireless network, there is no physical connection between members of the network. Instead, each member has its own wireless transmission range and a message can be transmitted to other nodes that are within the transmission range. This is called direct transmission. However, it is impossible to directly transmit a message from one node to another if they are not within the transmission range of each other. In this case, a message can be relayed through intermediate nodes. This collection of nodes that is used to transmit messages throughout the entire network is called a *virtual backbone*. The concept of a virtual backbone was proposed by Ephremides et al. [33]. If we assume that all members of a network have the same transmission range, then we can model this network using a unit disk graph. Virtual backbone has many advantages in terms of network management and routing. A *unit disk graph* is the intersection graph of a collection of unit disks in the Euclidean plane. For each disk, we form a vertex and join two vertices by an edge if the corresponding disks have non-empty intersection.

Many routing protocols based on connected dominating set (CDS) have been proposed [4, 7, 22, 28, 30, 57, 86, 95, 102, 103, 69, 100, 104, 97, 106, 96]. Since a message is transmitted through the virtual backbone, which is normally smaller than the size of the entire network, the routing search space can be reduced significantly. Small virtual backbone has the following advantages. First, it reduces the interference problem. Second, the smaller the CDS, the easier to maintain. Third, a small virtual backbone tends to provide more efficient routing and requires less control messages. Simonetti et al. [94] present an integer programming formulation, valid inequalities, and a branch-and-cut algorithm for minimum CDS problem. A brief summary of the formulation is shown in Appendix.

Mohammed et al. [80] shows that the chance of loss of data increases if a message is transmitted through a long path, and a CDS with small diameter consumes less energy compared to one with longer diameter. Diameter became a new critical factor in terms of the design of an algorithm for the construction of CDS. Numerical experiments in [80] show that their proposed algorithm generates a CDS with small diameter, but no theoretical proof is provided. Kim et al. [69] proposes two centralized CDS construction algorithms with approximation ratios for both CDS size and the diameter. The article also provides analysis of the algorithm given in [80].

Clique provides a very robust structure in the sense that all pairs of vertices are adjacent to each other and the deletion of any vertex does not destroy this structure. Recall that a dominating clique of a graph $G = (V, E)$ is a set $S \subset V$ of vertices such that S is a dominating set and the induced subgraph $G[S]$ is complete. We can think of the vertices in the dominating clique to be the communication core of a network. A dominating clique provides an extremely robust virtual backbone structure in wireless ad-hoc communication network in the sense that the core members can communicate directly and a message from any source to any destination can be transmitted within 3-hop. Minimum dominating clique problem seeks to find the smallest dominating clique and the size of the smallest dominating clique is called *dominating clique number* $\gamma_{cl}(G)$. In Figure 3.2, $S = \{2, 3, 5\}$ is a dominating clique. This problem is first introduced by Cozzens et al. [27] and it can be solved in polynomial time for strongly chordal graphs [72], undirected path graphs [72], and circle graphs [65]. However, in general, the minimum dominating clique problem is NP-hard [74]. Also, it is NP-complete to determine whether there exists a dominating clique in a graph [17, 18] and the problem remains NP-complete even when restricted to a perfect graph [18].

As mentioned above, a clique is overly restrictive and not all graphs have a dominating clique. As an alternative, we propose dominating s-club as a reliable

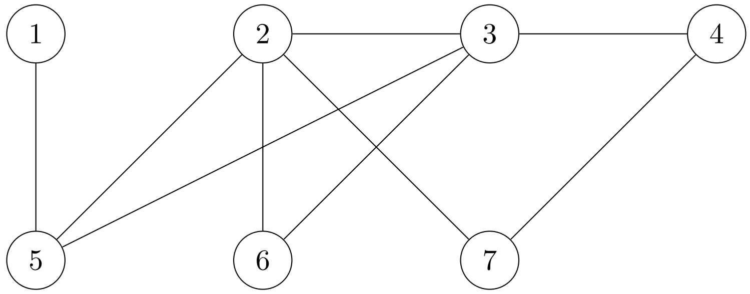


Figure 3.2: A graph with a dominating clique.

virtual backbone structure.

Definition 7. *(Dominating s-club (DsC)). Given a graph $G = (V, E)$ and a positive integer $s \geq 1$, a set $S \subset V$ of vertices is a dominating s-club if $|N(i) \cap S| \geq 1 \ \forall i \in V \setminus S$ and $diam(G[S]) \leq s$.*

Definition 8. *(Minimum dominating s-club (MDsC)). Given a graph $G = (V, E)$ and a positive integer $s \geq 1$, minimum dominating s-club problem seeks to find a DsC with minimum cardinality.*

The connected dominating set with bounded diameter has not been considered until recently. To the best of our knowledge, this is the first exact approach based on integer programming formulation for minimum dominating s-club problem for general $s \geq 1$. Note that if $s = 1$, then DsC is dominating clique. The size of the smallest dominating s-club is called *dominating s-club number* (or *DsC number*) and is denoted by $\gamma^s_{club}(G)$.

3.2 Existence of DsC

Not every graph has a dominating s-club. For instance, a cycle or a path of 6 vertices does not have D2C. We show that the problem of checking if a graph G has a DsC is NP-complete for any fixed $s \geq 1$ and also provide sufficient conditions for existence of a DsC.

Theorem 1. *The problem of checking the existence of a dominating s-club in a graph is NP-complete for any fixed integer $s \geq 1$.*

Proof. We first consider the case when s is odd. Let $x_1, x_2, \ldots, x_n$ be the set of variables and $C_1, \ldots, C_m$ be the set of clauses in an arbitrary instance $F = C_1 \wedge C_2 \wedge \ldots \wedge C_m$ of 3SAT. We construct a graph $G_F^s = (V_F^s, E_F^s)$ such that F is satisfiable if and only if G_F^s has a DsC. To construct G_F^s, let

$$V_F^s = V^{(0)} \cup V^{(1)} \cup V^{(2)} \cup V^{(3)} \text{ and } E_F^s = E^{(0)} \cup E^{(1)} \cup E^{(2)} \cup E^{(3)}, \text{ where}$$

$$\begin{aligned}
V^{(0)} &= \{a_i, \bar{a}_i | i = 1, 2, \ldots, n\}, \\
V^{(1)} &= \{b_i^{(r)}, \bar{b}_i^{(r)} | i = 1, 2, \ldots, n,\ r = 1, 2, \ldots, \frac{s+1}{2}\}, \\
V^{(2)} &= \{v_i^{(r)}, \bar{v}_i^{(r)} | i = 1, 2, \ldots, n,\ r = 1, 2, \ldots, \frac{s-1}{2}\}, \\
V^{(3)} &= \{c_j | j = 1, 2, \ldots, m\};
\end{aligned}$$

$$\begin{aligned}
E^{(0)} &= \{(a_i, a_{i'}), (a_i, \bar{a}_{i'}), (\bar{a}_i, \bar{a}_{i'}) | i, i' = 1, 2, \ldots, n, i \neq i'\}, \\
E^{(1)} &= \{(a_i, b_i^{(1)}), (\bar{a}_i, \bar{b}_i^{(1)}) | i = 1, 2, \ldots, n\} \\
&\quad \cup \{(b_i^{(r-1)}, b_i^{(r)}), (\bar{b}_i^{(r-1)}, \bar{b}_i^{(r)}) | i = 1, 2, \ldots, n, r = 2, \ldots, \frac{s+1}{2})\} \\
&\quad \cup \{(b_i^{(\frac{s-1}{2})}, \bar{b}_i^{(\frac{s-1}{2})}) | i = 1, 2, \ldots, n\}, \\
E^{(2)} &= \{(a_i, v_i^{(1)}), (\bar{a}_i, \bar{v}_i^{(1)}) | i = 1, 2, \ldots, n\} \\
&\quad \cup \{(v_i^{(r-1)}, v_i^{(r)}), (\bar{v}_i^{(r-1)}, \bar{v}_i^{(r)}) | i = 1, 2, \ldots, n, r = 2, \ldots, \frac{s-1}{2}\}, \\
E^{(3)} &= \{(v_i^{(\frac{s-1}{2})}, c_j) | j = 1, 2, \ldots, m, x_i \text{ is a literal in } C_j\} \\
&\quad \cup \{(\bar{v}_i^{(\frac{s-1}{2})}, c_j) | j = 1, 2, \ldots, m, \bar{x}_i \text{ is a literal in } C_j\}.
\end{aligned}$$

For $s = 1$ the set $V^{(2)}$ is empty, therefore in this case we replace vertices from $V^{(2)}$ in the definitions of $E^{(2)}$ and $E^{(3)}$ with the corresponding vertices from $V^{(0)}$. Then $E^{(2)}$ becomes a subset of $E^{(1)}$ and can be ignored. Figure 3.3 illustrates the proposed construction for $s = 5$.

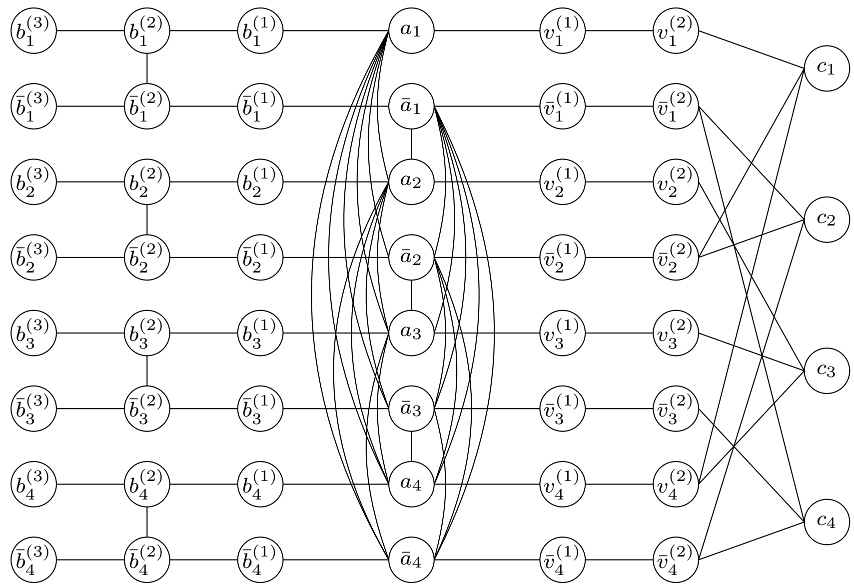


Figure 3.3: Illustration of the construction proposed in the proof of Theorem 1 for $s = 5$ and $F = (x_1 \vee \bar{x}_2 \vee x_4) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_4) \wedge (x_2 \vee x_3 \vee x_4) \wedge (\bar{x}_1 \vee \bar{x}_3 \vee \bar{x}_4)$.

We can check that F is satisfiable if and only if G_F^s has a DsC as follows. If F is satisfiable with a solution $x^* = (x_1^*, x_2^*, \ldots, x_n^*)$, then $D = V_F^s \backslash (V^{(3)} \cup \{b_i^{(\frac{s+1}{2})}, \bar{b}_i^{(\frac{s+1}{2})} | i = 1, \ldots, n\} \cup \{v_i^{(\frac{s-1}{2})} | i = 1, \ldots, n : x_i^* = 0\} \cup \{\bar{v}_i^{(\frac{s-1}{2})} | i = 1, \ldots, n : x_i^* = 1\})$ is a dominating set of diameter s in G_F^s. To establish the other direction, first note that no vertex from $V^{(3)}$ can be included in any dominating set of diameter at most s, since for $n > 3$ there are always vertices that are distance $s+2$ away from any vertex in $V^{(3)}$ and thus cannot be dominated by an s-club that contains a vertex from $V^{(3)}$. Also, recognize that for any $i \in \{1, \ldots, n\}$ the distance between $v_i^{(\frac{s-1}{2})}$ and $\bar{v}_i^{(\frac{s-1}{2})}$ is

$s+1$ in G_F^s; thus, only one of these vertices can belong to a DsC. Other than this, the distance between any pair of non-leaf vertices that are not in $V^{(3)}$ is at most s in G_F^s. In order for a DsC D to exist in G_F^s one must be able to pick one of the vertices $v_i^{(\frac{s-1}{2})}, \bar{v}_i^{(\frac{s-1}{2})}$ for each $i \in \{1, \ldots, n\}$ to include in D so that each vertex from $V^{(3)}$ is dominated by D. But this implies that F is satisfiable using the solution x^* with $x_i^* = 1$ if $v_i^{(\frac{s-1}{2})} \in D$ and $x_i^* = 0$ otherwise. The reduction is clearly polynomial.

When s is even, we use the construction for $s-1$ with the following changes. We replace each edge in $E^{(0)}$ with a path of length 2 by introducing a new "intermediate" vertex for each edge. Let $V^{(5)}$ be the set of all such intermediate vertices. We connect each pair of vertices in $V^{(5)}$ by an edge to make $V^{(5)}$ a clique. Also, to ensure that all vertices from $V^{(5)}$ belong to any CDS, we introduce a leaf vertex for each vertex in $V^{(5)}$ and add an edge connecting it to the corresponding vertex in $V^{(5)}$. Then, similarly to the case of odd s, the constructed graph has a DsC iff F is satisfiable. $\square$

The next theorem describes a class of graphs that are guaranteed to have a DsC. This result is a generalization of a theorem in [27], which proves that there is a dominating clique in a connected graph that is both C_5-free and P_5-free. We consider the case for $s=2$ first, then present the result for general $s \geq 2$.

Theorem 2. *If G is a connected graph with no induced P_6 and C_6, then G has a dominating 2-club.*

Proof. We use induction on n. This is clearly true for $n=1$. We assume that any connected graph of order n with no induced P_6 and C_6 has a dominating 2-club. Let G be a connected graph with $n+1$ vertices with no induced P_6 and C_6. Let $v \in V$ be a non-vertex-cut in G. Note that every connected graph has a vertex that is not a vertex-cut. Let $G' = G[V \setminus \{v\}]$. Since G' is connected and has no P_6 and C_6, it has a dominating 2-club S' from the induction hypothesis. If v has a neighbor in S', then S' is a dominating set for G.

Suppose that $N(v) \cap S' = \emptyset$. Since G is a connected graph, v is adjacent to a vertex $x \in (V - S')$. For notational simplicity, let Y_h, $h = 1, 2, 3$, be the set of vertices in $v \in S'$ such that $d_{G[\{x\} \cup S']}(x, v) = h$. Since S' is a 2-club, we can see that $S' = Y_1 \cup Y_2 \cup Y_3$. Let $S = \{x\} \cup (Y_1 \cup Y_2)$. We demonstrate that S is a dominating 2-club of G.

First, we demonstrate that S is a 2-club. Observe that any vertex in Y_1 is adjacent to x, while vertices in $Y_2 \cup Y_3$ are not adjacent to x. Any vertex in Y_2 is adjacent to at least one vertex in Y_1. Similarly, any vertex in Y_3 is adjacent to at least one vertex in Y_2 and not adjacent to any vertex in Y_1. Note that for any pair of vertices $i, j \in Y_1$, $d_{G[S]}(i, j) \leq 2$ since both i and j are adjacent to x in $G[S]$. Now we consider the cases where at least one of i, j is not in Y_1. To prove S is a 2-club, we show $Y_1 \cup Y_2$ form a 2-club by demonstrating that for any pair of vertices $i, j \in Y_1 \cup Y_2$ there is a shortest path between i and j with $d_{G[S']}(i, j) \leq 2$ not through any vertex in Y_3. Suppose there is a pair of vertices $i, j \in Y_1 \cup Y_2$ such that all shortest paths between them in $G[S']$ include a vertex from Y_3. We consider the following two possible cases. Case 1. One of i, j is in Y_1 and the other is in Y_2. Here, without loss of generality, assume $i \in Y_1$ and $j \in Y_2$. Case 2. $i, j \in Y_2$.

> Case 1. If $i \in Y_1$ and $j \in Y_2$, pick a shortest path between i and j through a vertex $w \in Y_3$, say $\{i, a, w, j\}$ where $a \in Y_2$. Then $\{v, x, i, a, w, j\}$ is an induced P_6, a contradiction.
>
> Case 2. If $i, j \in Y_2$, then clearly $(i, j) \notin E$ and no vertex in Y_1 is adjacent to both i and j. Consider a shortest path between x and i, say $\{x, a, i\}$ where $a \in Y_1$. Then $\{v, x, a, i, w, j\}$ is an induced P_6, a contradiction.

Thus, for any pair of vertices $i, j \in Y_1 \cup Y_2$, there is a shortest path between i and j not through any vertex in Y_3 meaning that $Y_1 \cup Y_2$ is a 2-club. Moreover, $d_{G[S]}(x, i) \leq 2$ for all $i \in S - \{x\}$ since i was selected to be within 2 hop from x. Thus, S is a 2-club.

Second, we demonstrate that S is a dominating set of G. Suppose that S is not a dominating set of G. Then there must be a vertex $u \in (V - S') - \{v\}$ that is not adjacent to any vertex in S. Note that any vertex in S' is dominated by S since it is either in S or has at least one neighbor in S. The fact that u was dominated by S' but not dominated by S implies that u was dominated by at least one vertex, say w, in Y_3, but not dominated by any vertex in $\{x\} \cup Y_1 \cup Y_2$ (i.e. u is not adjacent to any vertex in $\{x\} \cup Y_1 \cup Y_2$). If $(v, u) \notin E$, then $\{v, x, \ldots, w, u\}$ induces a path of 6 vertices, a contradiction. If $(v, u) \in E$, then $\{v, x, \ldots, w, u\}$ is an induced C_6 which is again a contradiction . □

Theorem 3. *Let $s \geq 2$ be an integer. If G is a connected graph with no induced P_{s+4} and no induced C_{s+4}, then G has a dominating s-club.*

Proof. We use induction on n. The proposition is clearly true for $n = 1$. We assume that any connected graph of order n with no induced P_{s+4} and no induced C_{s+4} has a dominating s-club. Let G be a connected graph on $n+1$ vertices with no induced P_{s+4} and no induced C_{s+4}. Let $v \in V$ be a vertex in G that is not a vertex-cut. Note that every connected graph has such a vertex. Let $G' = G[V \setminus \{v\}]$. Since G' is connected and has no induced P_{s+4} and no induced C_{s+4}, by the induction hypothesis, it has a dominating s-club S'. If v is adjacent to any vertex in S', then S' is a dominating s-club for G'.

Suppose that $N(v) \cap S' = \emptyset$. Since G is a connected graph, v is adjacent to a vertex $v' \in V \setminus S'$. For notational simplicity, let Y_h, $h = 0, 1, \ldots, s+1$, be the set of vertices in $S' \cup \{v'\}$ that are h hops away from v' in $G[S' \cup \{v'\}]$, i.e., $Y_h = \{v'' \in$

$S' \cup \{v'\} : d_{G[\{v'\}\cup S']}(v', v'') = h\}$. Note that $Y_0 = \{v'\}$ and $S' = \cup_{1\leq h\leq s+1} Y_h$. Let $S = \{v'\} \cup (\cup_{1\leq h\leq s} Y_h)$. We demonstrate that S is a dominating s-club of G.

First, we show that S is an s-club. Observe that each vertex $u \in Y_p$, $p \geq 2$, is adjacent to at least one vertex $u' \in Y_{p-1}$ and any vertex $u \in Y_q$, $q \geq 3$, is not adjacent to any vertex $u' \in Y_p$, $p \leq q-2$, in $G[S']$, since otherwise u would have been included in some Y_r with $r < q$. Note that for any $u, u' \in S'$ we have $d_{G[S']}(u, u') \leq s$ since S' is an s-club, and for any $u \in S$ we have $d_{G[S]}(v', u) \leq s$ by the construction of S. Thus, to prove S is an s-club, it is sufficient to show that for any pair of vertices $u, u' \in \cup_{1\leq h\leq s} Y_h$ there is a path of length at most s between u and u' in $G[S']$ that does not include a vertex from Y_{s+1}. Suppose there is a pair of vertices $u \in Y_p$, $u' \in Y_q$, where $p \leq q$, such that all paths of length at most s, including all shortest paths, between them in $G[S']$ include a vertex from Y_{s+1}. Consider one such shortest path $P_{uu'}$ passing through $u'' \in Y_{s+1}$. Then $P_{uu'} = P_{uu''} \cup P_{u''u'}$, where $P_{uu''}$ and $P_{u''u'}$ are shortest paths between u, u'' and u'', u', respectively. Obviously, $P_{u''u'}$ consists of at least two vertices, u'' and u'. Next, consider a shortest path P_{vu} from v to u in $G[S \cup \{v\}]$. Since v' is the only neighbor of v in S and $u \in Y_p$ implies that $d_{G[S]}(v', u) = p$, the length of P_{vu} is $p + 1$. Note that in general, the path $P_{uu'}$ may contain vertices from Y_r with $r \leq p$. Let r^* be the smallest r such that Y_r contains a vertex from $P_{uu'}$, and let $u^* \in Y_{r^*}$ be the vertex that is the farthest from u among the vertices in $P_{uu'}$ that belong to Y_{r^*}. Denote by P_{vu^*} a shortest path between v and u^* in $G[\{v\} \cup S]$ and by $P_{u^*u'}$ a shortest path between u^* and u' in $G[S]$. Then $P_{vu^*} \cup P_{u^*u'}$ is a path containing at least $s + 4$ vertices and we obtain a contradiction to the assumption that G' has no induced P_{s+4}.

Second, we demonstrate that S is a dominating set of G. Suppose it is not. Then there must be a vertex $u \in V \setminus (S' \cup \{v\})$ that is not adjacent to any vertex in S. Note that any vertex in S' is dominated by S since $S' \setminus S = Y_{s+1}$ and each vertex in Y_{s+1} has a neighbor in $Y_s \subset S$. The fact that u is dominated by S' but is not dominated by S implies that u is dominated by at least one vertex, say w, in Y_{s+1}, but is not dominated by any vertex in $\cup_{1 \leq h \leq s} Y_h \cup \{v'\}$. Let $P_{v'w}$ be a shortest path between v' and w in $G[\{v'\} \cup S']$. Since $w \in Y_{s+1}$, $P_{v'w}$ consists of $s+2$ vertices. Then, depending on whether $(v,u) \in E$, $\{v\} \cup P_{v'w} \cup \{u\}$ induces P_{s+4} or C_{s+4}, a contradiction. Thus, S must be a dominating set of G. □

We can easily see that the converse of Theorem 3 does not hold. For instance, consider a graph created by adding a node v to P_{s+4}, with edges between v and every other vertex. Clearly v is a DsC; however, the graph has an induced P_{s+4}. It is easy easy to see that for any $s \geq diam(G)$, there is a DsC in $G = (V, E)$ since V is a DsC.

Theorem 4. *For a graph $G = (V, E)$ with $diam(G) \geq 4$, there is no DsC in G for any $s < diam(G) - 2$.*

Proof. Consider two vertices $v, v' \in V$ such that $d_G(v, v') = diam(G)$. A DsC must include one vertex $u \in N(v)$ and one vertex $u' \in N(v')$, since otherwise it is not connected. However, for any $S \subseteq V$, $d_{G[S]}(u, u') \geq d_G(u, u') \geq diam(G) - 2 > s$, indicating that u and u' cannot be in the same s-club. □

3.3 Complexity of the MDsC problem

We consider the decision version of the MDsC problem, Dominating s-Club (DsC) as follows. 'Given a graph $G = (V, E)$ and a positive integer s, does there exist a DsC of size $\leq k$ in G?' We prove that for each fixed positive integer s, DsC is NP-complete, even if restricted to graphs for which a DsC is known to exist.

Theorem 5. DsC *is NP-complete for any fixed positive integer s, even if restricted to graphs in which a dominating clique exists.*

Proof. First we observe that the problem is clearly in the class NP, then we reduce the classical NP-complete Vertex Cover (VC) problem to DsC. Let a graph $G = (V, E)$ and a positive integer s be an instance of VC, which asks if there exists a subset of at most s vertices C in G such that each edge has at least one endpoint in C.

We construct an instance $G' = (V', E')$ of DsC as follows. Let $V' = V \cup W$, where $W = \{w_{uv} : (u, v) \in E\}$. Each vertex w_{uv} in W is joined by an edge to both u and v. We also add an edge between every pair of vertices in V' that represent the elements of V. Thus, $E' = \{(w_{uv}, u) : (u, v) \in E\} \cup \{(u, v) : u, v \in V\}$. Obviously, V is a dominating clique in G'.

We show that G has a vertex cover of size $\leq k$ if and only if G' has a DsC of size $\leq k$. Let $C \subseteq V$ be a vertex cover in G. Then C is a dominating clique in G', and hence a DsC for any s in G'. Now, let D be a DsC in G'. If $D \cap W \neq \emptyset$, replacing each vertex of W in D with one of its two neighbors from V results in a dominating clique C of size $|C| \leq |D|$ in G'. Clearly C is a vertex cover in G. $\square$

3.4 Bounds on the DsC number

Recall that the DsC number of G is denoted by $\gamma^s_{club}(G)$. Similarly, let $\gamma(G)$ be the domination number, $\gamma_t(G)$ be the total domination number, $\gamma_c(G)$ be the connected domination number, and $\gamma_{cl}(G)$ be the dominating clique number of G.

Theorem 6. *If a graph G has a dominating clique and $\gamma(G) \geq 2$, then for $s' \geq s \geq 2$ the following inequalities hold:*

$$\gamma(G) \leq \gamma_t(G) \leq \gamma_c(G) \leq \gamma^{s'}_{club}(G) \leq \gamma^s_{club}(G) \leq \gamma_{cl}(G). \tag{3.1}$$

Proof. Since a dominating clique is a dominating s-club, it follows that for $s \geq 2$

the size of a smallest dominating s-club is at most the size of a smallest dominating clique. For $s' \geq s \geq 2$, we can easily see that every dominating s-club is also a dominating s'-club, but the converse is not true. Similarly, a dominating s'-club is also a CDS, but the converse is not true. The relations between the size of the minimum dominating set, minimum total dominating set, and minimum CDS can be found in [52]. $\square$

3.5 IP formulation

We use formulation ideas proposed by [98] for the maximum s-club problem. Let the set of vertices be labeled $V = \{1, 2, \dots, n\}$. For $D \subseteq V$ the vector $x \in \{0,1\}^n$ such that $x_i = 1$ iff $i \in D$ is called the characteristic vector of D. Then the MDsC problem can be formulated as the following IP.

$$\text{minimize} \sum_{i \in V} x_i \tag{3.2}$$

$$\text{subject to} \sum_{j \in N[i]} x_j \geq 1, \quad i \in V; \tag{3.3}$$

$$\sum_{r=2}^{s} y_{ij}^{(r)} \geq x_i + x_j - 1, \quad i \in V, j \in V \setminus N[i]; \tag{3.4}$$

$$y_{ij}^{(2)} \leq x_i, \quad y_{ij}^{(2)} \leq x_j, \quad y_{ij}^{(2)} \leq \sum_{p \in N(i) \cap N(j)} x_p, \quad j > i, \ i, j \in V; \tag{3.5}$$

$$y_{ij}^{(2)} \geq C \sum_{p \in N(i) \cap N(j)} x_p + (x_i + x_j - 2), \quad j > i, \ i, j \in V; \tag{3.6}$$

$$y_{ij}^{(r)} \leq x_i, \quad y_{ij}^{(r)} \leq \sum_{p \in N(i)} y_{pj}^{(r-1)}, r = 3, \dots, s; \quad j > i, \ i, j \in V; \tag{3.7}$$

$$y_{ij}^{(r)} \geq C \sum_{p \in N(i)} y_{pj}^{(r-1)} + (x_i - 1), \quad j > i, \ i, j \in V; \tag{3.8}$$

$$x_i \in \{0,1\}, \quad y_{ij}^{(r)} \in \{0,1\}, \quad i, j \in V, r \in \{2, \dots, s\}. \tag{3.9}$$

In this formulation, the n-vector x of 0-1 variables is the characteristic vector of the DsC. The binary decision variable $y_{ij}^{(r)}$ takes a value of 1 if and only if there exists a path of length r from vertex i to vertex j in the DsC. The objective is to find the dominating s-club with minimum cardinality. Note that C is a constant and it can be set to $C = \frac{1}{n}$. As proposed in [98], setting $C = |N(i) \cap N(j)|^{-1}$ results in a tighter formulation. However, it causes a problem when $|N(i) \cap N(j)| = 0$. Constraint (3.3) is a dominating constraint which ensures that each vertex is dominated. Proposition 1 describes that this closed neighborhood constraint can be replaced with open neighborhood inequality $\sum_{j \in N(i)} x_j \geq 1 \; i \in V$ when there is no single vertex that dominates V. Constraint (3.4) ensures that the distance between the dominating vertices $i, j \in V$ is at most s. Constraints (3.5) and (3.6) are for paths of length 2, whereas constraints (3.7) and (3.8) are for paths of length more than 2. We refer the reader to [98] for more detail on the s-club (diameter) constraints.

3.6 Valid inequalities and variable fixing

In this section, we consider some valid inequalities and variable fixing techniques for DsC. As mentioned in Theorem 6, a feasible solution to MDsC problem is also a feasible solution to MCDS problem. This implies that the convex hull of feasible solutions of MDsC problem is a subset of that of MCDS problem. We conclude that an inequality which is valid for CDS polytope is valid for DsC polytope. A subset $C \subset V$ of vertices is called a vertex-cut if $G[V \setminus C]$ is disconnected.

Proposition 1. *The following are valid for any characteristic vector x of a DsC or a CDS.*

- **Vertex-cut inequalities**. *For any vertex-cut C, we have*

$$\sum_{i \in C} x_i \geq 1. \tag{3.10}$$

- **Open neighborhood inequalities.** *Suppose that $\gamma(G) \geq 2$, then*

$$\sum_{j \in N(i)} x_j \geq 1 \quad i \in V. \tag{3.11}$$

- **Vertex-cut fixing**. *If $i \in V$ is a vertex-cut, then $x_i = 1$.*

Proof. The vertex-cut inequalities have been established by [105]. Open neighborhood inequalities follow from the vertex-cut inequalities because $N(i)$ is a vertex-cut. Vertex-cut fixing is given by [78]. It also follows from the vertex-cut inequalities by letting $C = \{i\}$. □

Remark 1. *Note that the open neighborhood inequalities for DsC and CDS subsume the classical domination constraints.*

Proposition 2. *The following statement is valid for DsC and CDS. If there are $J \subset V$ and $I \subseteq V \setminus J$ such that $\forall i \in I\ \exists j \in J$ with $N[i] \subseteq N[j]$ or $N(i) \subseteq N(j)$, then there is an optimal solution with $x_i = 0\ \forall i \in I$.*

Proof. For closed neighborhood inclusion based variable fixing for MDS and MCDS problem, refer to [2] and [78], respectively. To prove the statement for the MDsC problem, consider two subsets of vertices I and J satisfying the closed neighborhood inclusion. Assume that for some $i \in I$, $x_i = 1$ in some feasible solution. For an arbitrary $j \in J$ with $N[i] \subseteq N[j]$, x can be changed to have a feasible solution with the same objective value by setting $x_i = 0$ and $x_j = 1$ if $x_j \neq 1$. Now, every vertex that was dominated by i is dominated by j. It is easy to check that the diameter has not changed. The same procedure can be applied to obtain a feasible solution with $x_i = 0 \; \forall i \in I$. The proof for open neighborhood inclusion is similar, and it is omitted. $\square$

Note that the collection of all minimal vertex-cut inequalities along with binary constraints give a proper mathematical programming formulation for MCDS problem [105]. We generalize this approach to develop a formulation for minimum k-connected m-dominating set in section 4. The valid inequalities established in the previous proposition work both for CDS and DsC. The next proposition establishes stronger inequalities that are only valid for DsC.

Proposition 3. *Assume G has a DsC, and let x be the characteristic vector of an arbitrary DsC. Then the following statements hold.*

- **Diameter-critical set inequalities.** *If there is a set $S \subseteq V$ with $diam(G[V \setminus S]) \geq s + 3$, then*

$$\sum_{i \in S} x_i \geq 1. \tag{3.12}$$

- **Diameter-critical vertex fixing.** *If there exists a vertex $i \in V$ with $diam(G[V \setminus \{i\}]) \geq s + 3$, then $x_i = 1$.*

Proof. Suppose G has a DsC $D' \subseteq V \setminus S$. Since D' dominates G, D' must also dominate $G[V \setminus S]$, which means that D' is a DsC for $G[V \setminus S]$. By Theorem 4, $G[V \setminus S]$ has no DsC for $s < diam(G[V \setminus S]) - 2$. However, $s \leq diam(G[V \setminus S]) - 3$, a contradiction. Similar proof holds for diameter-critical vertex fixing. $\square$

Remark 2. *Note that the diameter-critical set inequalities subsume the vertex-cut inequalities.*

3.7 Experimental results

All computational experiments were conducted on *Dell Precision WorkStation T7500*® computers, each with eight 2.40 GHz Intel Xeon® processors, and 12 GB RAM. The solver used was ILOG CPLEX 12.1®.

In a graph $G = (V, E)$ which models a wireless communication network, a pair of nodes can communicate directly when they are within the transmission range of each other. Assuming that every node has the same transmission range, we can model the network using a unit disk graph. In a unit disk graph, there is an edge between two nodes if the distance between them is less than unit distance (i.e. the center of one node is within the unit disk of the other). Consider a 2-dimensional square box of certain size. Within the box, we create a certain number of points randomly. For each distinct pair of points, connect them using an edge if the distance between them is less than a specified distance. The specific values used for size of box, the number of points, and the distance criteria are the same as used in [19].

Before solving MDsC problem using CPLEX, we check if there is a universal vertex (a vertex $i \in V$ such that $N[i] = V$) or a universal edge (an edge $(i, j) \in E$ such that $N[i] \cup N[j] = V$) since if there exist one of these, then the problem is solved. If neither of these exist, then a heuristic algorithm from [19] is applied to obtain a CDS $S \subset V$. If $diam(G[S]) \leq s$, then S is used as the initial feasible solution for the MDsC problem.

Variable fixing based on open and closed neighborhood inclusion is applied in the following way. If the closed (open) neighborhood of a vertex i is a strict subset of the closed (open) neighborhood of a vertex j, then we set $x_i = 0$. On the other hand, if the closed (open) neighborhood of i is equal to the closed (open) neighborhood of j, then we set $x_{min\{i,j\}} = 0$. Note that by proposition 2 there is an optimal solution after this procedure.

The preliminary set of experiments are performed to examine whether the valid inequalities and variable fixing techniques are useful for solving MDsC problem. For this purpose, several random unit disk graphs are created with 100 vertices, with the length of the box size at 100 and the unit disk radius ranging between 20 and 50, with an increment of 5. Table 3.1 shows the characteristics of the created graphs for each radius such as the edge density $\rho(G)$, diameter $(diam(G))$, and the size of a minimum DsC $(\gamma^s_{club}(G))$ for $s = diam(G) - 2$. The table also includes the problem solving time in seconds for these preliminary experiments. The column "Basic" indicates the results for the basic version of formulation (3.3)–(3.9) and other labels in the table correspond to the following variations of the IP approach.

- Apply diameter-critical vertex fixing to the basic implementation ("DCF");
- Apply variable fixing based on neighborhood inclusion to the basic implementation ("NIF");
- Replace the standard closed neighborhood based domination constraints (3.4) with the open neighborhood inequalities (3.11) ("ONI");
- Replace the standard closed neighborhood based domination constraints (3.4) with the open neighborhood inequalities (3.11), and apply diameter-critical vertex fixing, and variable fixing based on neighborhood inclusion to the basic implementation ("All").

In Table 3.1, "$> 24hrs$" indicates that the corresponding instance is not terminated within 24 hours. As these preliminary test results show, it is typically most beneficial to include all of the proposed enhancement techniques to the basic implementation. Therefore, for the remaining experiments we apply all of those methods and the results are presented in Tables 3.2–3.4. Three lowest values of s are used: $s = diam(G) - 2$, $s = diam(G) - 1$, and $s = diam(G)$. Time limit is set as 3600 seconds. For instances that are not solved within the time limit, the size of the best found DsC is reported. The column labeled "gap" shows the optimality gap reported by CPLEX for each run.

Table 3.1: Results of the experiments with variations of the IP approach for $s = diam(G) - 2$ on a set of 100-vertex random unit disk graphs with the length of the box side set at 100.

radius	$\rho(G)$	$diam(G)$	$\gamma^s_{club}(G)$	CPU time (seconds)				
				Basic	DCF	NIF	ONI	All
20	0.10	9	19	> 24 hrs	> 24 hrs	14534.8	> 24 hrs	5344.87
25	0.16	6	12	22929.9	18521.6	335.798	10840	206.437
30	0.24	5	9	11.185	8.471	2.019	9.438	2.808
35	0.26	4	6	3.323	1.997	1.23	3.463	1.872
40	0.33	4	5	4.821	4.836	1.698	4.992	2.512
45	0.46	4	4	4.134	4.072	1.3	4.977	2.324
50	0.48	3	3	2.762	2.761	1.465	2.808	2.683

Tables 3.2 and 3.3 are the results of experiments for 100-vertice and 150-vertice instances of random unit disk graphs, respectively. A blank entry indicates that $s < 1$. The first column of these tables represents the dimensions of the box used to create the graphs. The remaining column notations have the same meaning as in Table 3.1. The set of 100×100 instances in Table 3.2 is the same as the ones used in Table 3.1.

Table 3.4 shows the experiment results on a set of MCDS problem instances from [78] and [94]. Note that the blank entries indicate that $s < 1$. The symbol † represents that CPLEX does not solve the linear programming relaxation within the time limit with its default setting. For these, we apply interior point method to find

Table 3.2: Results for random unit disk graphs with $n = 100$.

				$s = diam(G)$			$s = diam(G) - 1$			$s = diam(G) - 2$		
dimensions	radius	$\rho(G)$	$diam(G)$	$\gamma^s_{club}(G)$	time	gap	$\gamma^s_{club}(G)$	time	gap	$\gamma^s_{club}(G)$	time	gap
100×100	20	0.10	9	≤ 20	> 3600	0.34	≤ 21	> 3600	0.37	19	> 3600	0.13
	25	0.16	6	≤ 11	> 3600	0.27	≤ 11	> 3600	0.24	12	206.4	0
	30	0.24	5	8	2538.0	0	8	406.0	0	9	2.8	0
	35	0.26	4	6	102.0	0	6	4.4	0	6	1.9	0
	40	0.33	4	5	4.2	0	5	4.1	0	5	2.5	0
	45	0.46	4	4	5.6	0	4	3.8	0	4	2.3	0
	50	0.48	3	3	3.8	0	3	2.2	0	3	2.7	0
120×120	20	0.08	12	≤ 25	> 3600	0.29	≤ 26	> 3600	0.26	≤ 24	> 3600	0.09
	25	0.10	9	≤ 21	> 3600	0.37	≤ 22	> 3600	0.36	≤ 20	> 3600	0.09
	30	0.14	7	≤ 12	> 3600	0.25	≤ 12	> 3600	0.25	12	230.2	0
	35	0.19	5	≤ 10	> 3600	0.20	9	732.5	0	11	2.4	0
	40	0.24	5	≤ 7	> 3600	0.14	7	979.5	0	7	13.0	0
	45	0.35	4	5	107.9	0	5	3.6	0	5	2.6	0
	50	0.39	4	4	4.3	0	4	3.7	0	4	2.7	0
140×140	30	0.13	7	≤ 15	> 3600	0.27	≤ 15	> 3600	0.24	15	715.5	0
	35	0.16	6	≤ 13	> 3600	0.40	≤ 11	> 3600	0.24	12	79.7	0
	40	0.19	6	≤ 9	> 3600	0.22	≤ 10	> 3600	0.30	9	4.1	0
	45	0.26	5	7	3452.8	0	7	1097.0	0	7	15.7	0
	50	0.32	4	5	418.5	0	5	4.7	0	5	2.3	0
	55	0.37	4	5	97.8	0	5	3.8	0	5	2.9	0
	60	0.40	3	4	3.1	0	4	2.8	0	4	2.4	0
160×160	30	0.09	11	≤ 22	> 3600	0.36	≤ 22	> 3600	0.34	≤ 21	> 3600	0.22
	35	0.11	8	≤ 20	> 3600	0.44	≤ 60	> 3600	0.81	≤ 17	> 3600	0.16
	40	0.16	6	≤ 12	> 3600	0.29	≤ 11	> 3600	0.17	12	245.6	0
	45	0.21	6	≤ 12	> 3600	0.42	≤ 66	> 3600	0.89	10	2867.5	0
	50	0.22	5	6	184.6	0	6	152.6	0	6	3.1	0
	55	0.27	5	6	196.7	0	6	1157.4	0	6	18.0	0
	60	0.32	4	5	4.4	0	5	4.1	0	5	2.7	0

an appropriate gap. An asterisk $*$ indicates that no solution was found in the time limit, and the gap was not reported. The symbol *inf* indicates that the instance is infeasible.

Notice that typically the best results in terms of solution time are observed when s is small and the edge density of a graph is high. For 100-vertex instances, optimal solution is reached within time limit when the edge density is at least 25%. Similarly, for 150-vertex instances, optimal solutions are found within time limit when the edge density is at least 30%. In general, sparse instances are more difficult to solve. This is consistent to the result reported for MCDS problem [94] and for the maximum s-club problem [98].

Table 3.3: Results for random unit disk graphs with $n = 150$. A blank entry indicates that $s < 1$.

				$s = diam(G)$			$s = diam(G) - 1$			$s = diam(G) - 2$		
dimensions	radius	$\rho(G)$	$diam(G)$	$\gamma^s_{club}(G)$	time	gap	$\gamma^s_{club}(G)$	time	gap	$\gamma^s_{club}(G)$	time	gap
120×120	50	0.39	4	4	19.1	0	4	15.1	0	4	11.4	0
	55	0.44	3	4	14.2	0	4	8.0	0	4	10.7	0
	60	0.51	3	3	17.1	0	3	14.9	0	3	12.3	0
	65	0.52	3	3	17.0	0	3	9.4	0	3	12.4	0
	70	0.61	3	2	0.0	0	2	0.0	0	2	0.0	0
	75	0.63	3	2	0.0	0	2	0.0	0	2	0.0	0
	80	0.72	2	2	0.0	0	2	0.0	0			
140×140	50	0.30	4	≤ 7	> 3600	0.29	5	16.6	0	6	9.0	0
	55	0.34	4	4	17.4	0	4	11.2	0	5	10.9	0
	60	0.35	4	4	15.3	0	4	14.0	0	4	10.1	0
	65	0.43	4	4	19.1	0	4	13.4	0	4	12.5	0
	70	0.47	3	3	14.9	0	3	8.5	0	4	11.3	0
	75	0.53	3	2	0.0	0	2	0.0	0	2	0.0	0
	80	0.57	3	2	0.0	0	2	0.0	0	2	0.0	0
160×160	50	0.24	5	≤ 9	> 3600	0.33	≤ 8	> 3600	0.25	7	1552.9	0
	55	0.29	4	≤ 6	> 3600	0.17	6	241.6	0	6	8.4	0
	60	0.29	4	5	196.6	0	5	135.6	0	5	8.9	0
	65	0.35	4	4	138.9	0	4	16.5	0	5	12.6	0
	70	0.43	4	4	20.8	0	4	18.5	0	4	12.7	0
	75	0.42	3	3	12.7	0	3	7.3	0	4	10.4	0
	80	0.51	3	3	17.1	0	3	9.4	0	3	12.2	0
180×180	50	0.18	6	≤ 13	> 3600	0.46	≤ 13	> 3600	0.46	≤ 11	> 3600	0.36
	55	0.21	5	≤ 9	> 3600	0.33	≤ 10	> 3600	0.4	8	8.6	0
	60	0.24	5	≤ 9	> 3600	0.39	≤ 8	> 3600	0.33	7	296.1	0
	65	0.26	4	5	1382.5	0	5	69.6	0	6	8.5	0
	70	0.32	4	5	517.7	0	5	60.8	0	6	10.7	0
	75	0.34	4	4	13.4	0	4	9.9	0	5	10.3	0
	80	0.38	4	4	17.4	0	4	16.2	0	4	11.7	0

Table 3.4: Test results for instances from [78] and [94].

			$s = diam(G)$			$s = diam(G) - 1$			$s = diam(G) - 2$		
n	$\rho(G)$	$diam(G)$	$\gamma^s_{club}(G)$	time	gap	$\gamma^s_{club}(G)$	time	gap	$\gamma^s_{club}(G)$	time	gap
30	0.10	8	15	1.5	0	15	0.7	0	*inf*	0.4	0
	0.20	5	7	6.2	0	7	0.6	0	8	0.7	0
	0.30	3	4	1.0	0	5	0.6	0	*inf*	0.3	0
	0.50	2	3	0.6	0	3	0.3	0			
	0.70	2	2	0.0	0	2	0.0	0			
50	0.05	14	≤ 31	> 3600	0.05	32	1598.0	0	32	182.7	0
	0.10	5	13	487.1	0	14	33.2	0	19	1.3	0
	0.20	3	7	30.8	0	7	1.2	0	*inf*	0.3	0
	0.30	3	5	1.8	0	5	2.2	0	*inf*	0.4	0
	0.50	2	3	1.2	0	3	0.4	0			
	0.70	2	2	0.0	0	2	0.0	0			
70	0.05	8	≤ 32	> 3600	0.25	≤ 29	> 3600	0.10	32	249.8	0
	0.10	4	13	1050.2	0	17	55.5	0	*inf*	3.4	0
	0.20	3	7	162.1	0	8	8.6	0	*inf*	0.6	0
	0.30	3	5	436.4	0	5	6.1	0	*inf*	0.8	0
	0.50	2	3	2.6	0	3	1.0	0			
	0.70	2	2	0.0	0	2	0.0	0			
100	0.05	5	≤ 34	> 3600	0.39	40	3316.2	0	*inf*	8.8	0
	0.10	4	≤ 15	> 3600	0.24	≤ 18	> 3600	0.34	*inf*	15.7	0
	0.20	3	≤ 10	> 3600	0.42	9	305.6	0	*inf*	1.5	0
	0.30	2	6	48.2	0	*inf*	2.9	0			
	0.50	2	4	16.6	0	4	3.2	0			
	0.70	2	3	26.1	0	3	4.1	0			
120	0.05	6	≤ 30	> 3600	$0.25^{\dagger}$	≤ 34	> 3600	0.34	≤ 41	> 3600	0.37
	0.10	3	≤ 20	> 3600	0.43	*inf*	43.8	0	*inf*	24.2	0
	0.20	3	≤ 10	> 3600	0.41	≤ 10	> 3600	0.23	*inf*	2.7	0
	0.30	3	≤ 8	> 3600	0.50	6	280.0	0	*inf*	4.8	0
	0.50	2	4	49.2	0	4	6.1	0			
	0.70	2	3	645.2	0	3	7.7	0			
150	0.05	5	≤ 37	> 3600	$0.42^{\dagger}$	≤ 39	> 3600	0.44	*inf*	66.0	0
	0.10	3	≤ 22	> 3600	0.47	*inf*	260.5	0	*inf*	97.2	0
	0.20	3	≤ 13	> 3600	0.56	≤ 12	> 3600	0.45	*inf*	6.0	0
	0.30	2	6	841.9	0	*inf*	12.9	0			
	0.50	2	4	144.2	0	4	13.2	0			
	0.70	2	3	2187.2	0	3	17.9	0			
200	0.05	4	≤ 46	> 3600	$0.51^{\dagger}$	$*$	> 3600	$*$	*inf*	224.6	0
	0.10	3	≤ 25	> 3600	0.58	$*$	> 3600	$*$	*inf*	399.0	0
	0.20	3	≤ 19	> 3600	0.73	≤ 13	> 3600	0.57	*inf*	18.5	0
	0.30	2	≤ 7	> 3600	0.48	*inf*	72.2	0			
	0.50	2	4	1635.9	0	4	41.5	0			
	0.70	2	≤ 4	> 3600	0.50	3	53.8	0			

4. FAULT-TOLERANT CONNECTED DOMINATING SET

In this chapter, we study the minimum k-connected m-dominating set problem, which generalizes the well-studied minimum connected dominating set problem. Despite its popularity, no exact approach has been applied to solve the minimum k-connected m-dominating set problem. We present some nice structural characteristics of the problem, especially when $m \geq k$. Robustness is a desirable property, but we show that it is not free. In fact, accomplishing robustness requires a strictly more-costly solution. Three different mathematical programming formulations based on vertex-cuts are presented depending on whether $m < k$, $m = k$, or $m > k$. We present some fundamental study results of the corresponding polytope. The separation problem for vertex-cut inequalities is a weighted vertex-connectivity problem and it can be solved in polynomial time. We generalize vertex-cut and vertex-cut inequalities by introducing r-robust vertex-cut and r-robust vertex-cut inequalities. For numerical experiment, we consider a lazy-constraint approach and the test results show that the proposed method compares favorably with existing approaches for minimum 1-connected 1-dominating set (or minimum connected dominating set) problem. Test results for $k = m = 2, 3,$ and 4 are presented as well.

4.1 Definitions and previous work

A *vertex-cut* $C \subset V$ of a connected graph G is a set of vertices such that $G[V \setminus C]$ is disconnected. The *vertex-connectivity* $\kappa(G)$ (or simply, *connectivity*) for a graph G, where G is not a complete graph, is the size of a minimal vertex-cut. A graph is *k-vertex-connected* (or, simply *k-connected*) if its vertex connectivity is k or greater.

[*]Parts of this section are based on the working paper Austin Buchanan, Je Sang Sung, Sergiy Butenko, Eduardo L. Pasiliao: An integer programming approach for k-connected d-dominating sets.

Note that a graph G is k-connected when there does not exist a subset $C \subset V$ of vertices with $|C| = k - 1$ such that $G[V \setminus C]$ is disconnected. A complete graph with n vertices does not have a vertex-cut at all. By convention, the connectivity of a complete graph of n vertices is $n - 1$, i.e. $\kappa(K_n) = n - 1$. Alternatively, we can say that a connected graph G is k-vertex-connected if there exist k vertex-disjoint paths between every pair of distinct vertices. This definition also applies to a complete graph.

In wireless ad-hoc network, a connected dominating set (CDS) is proposed to serve as a virtual backbone. When a message from a source node cannot be directly passed to the destination node, the message can be relayed through the intermediate nodes. A CDS can serve as these intermediate nodes. Finding the minimum CDS has drawn significant attention. The advantage of a minimum CDS is that it requires the smallest number of functioning units to form a virtual backbone. This simple structure, however, can fail if a node in a virtual backbone fails to work. We study robust virtual backbone structure by considering k-connected m-dominating set. Note that a k-connected k-dominating set remains a CDS if fewer than k vertices fail. The parameter m in m-dominating is usually reserved to represent the number of edges in a graph, so we use d instead.

Definition 9. *(k-connected d-dominating set). Given a connected graph $G = (V, E)$ with integers $k, d \geq 1$, a set $S \subset V$ is k-connected d-dominating set (k-d-CDS) if $\kappa(G[S]) \geq k$ and $|N(i) \cap S| \geq d \; \forall i \in V \setminus S$.*

Definition 10. *(Minimum k-d-CDS problem). Given a connected graph $G = (V, E)$ with integers $k, d \geq 1$, minimum k-connected d-dominating set (Mk-d-CDS) problem seeks to find the smallest k-d-CDS, or decide that none exist.*

We focus on the minimum k-d-CDS problem in this chapter. It is easy to see that a k-d-CDS generalizes connected dominating set ($k = d = 1$), and the minimum k-d-CDS problem generalizes the classical MCDS problem. Most of the results presented here apply to the case $d \geq k$. For instance, we show that every superset of a k-d-CDS is also a k-d-CDS when $d \geq k$, while this does not hold for $d < k$. In Figure 4.1, for $k = 4$ and $d = 3$, $\{1, 2, 3, 4, 5\}$ is 4-3-CDS, but the entire vertex set is not.

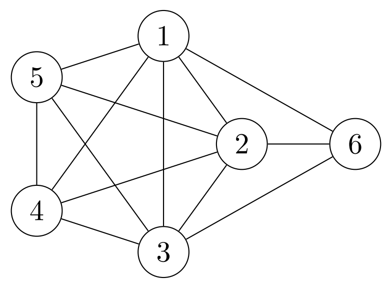


Figure 4.1: When $k > d$, a superset of k-d-CDS may not necessarily be a k-d-CDS.

A variety of approaches have been applied to solve the MCDS problem and closely related problem called maximum leaf spanning tree problem, including exact approaches [39, 78, 94, 82, 34, 40, 41], approximation algorithms [48, 76], and heuristic algorithms [19, 8]. There exist polynomial-time approximation schemes (PTAS) for unit-disk graphs [23, 58] and for unit-ball graphs [107]. Dai and Wu [29] propose a k-k-CDS as a virtual backbone structure and introduce several algorithms. Thai et al. [97] study a general fault tolerant CDS and propose two approximation algorithms for 1-d-CDS and k-k-CDS in heterogeneous networks. Wu and Li [104] propose a distributed algorithm with low message complexity to construct a k-d-CDS for general k and d. Wu and Li [104] also show that their proposed algorithm has a constant factor approximation ratio when the maximum node degree is a constant. Wang et al. [100] propose a constant factor approximation algorithm of 2-connected virtual backbone.

4.2 Properties of k-d-CDS for $d \geq k$

Property 1. *Let G be a k-connected graph for $k \geq 1$, then $\delta(G) \geq k$.*

Proof. Suppose not. Then, there is a vertex $v \in V$ such that $deg_G(v) = t$, $t \leq k-1$. Let $N(v) = \{u_1, \ldots, u_t\}$. Then removal of $N(v)$ isolates vertex v, a contradiction. $\square$

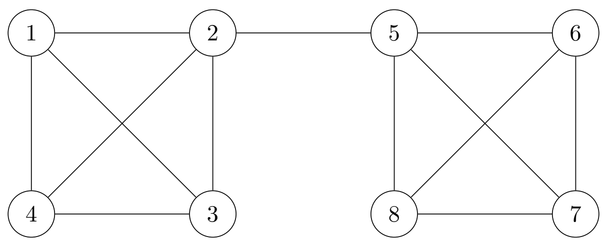


Figure 4.2: $\delta(G) \geq k$ does not imply $\kappa(G) \geq k$.

However, it is easy to see that the reverse of Property 1 does not hold. For instance, in Figure 4.2, $\delta(G) \geq 3$ but the graph is not 3-connected. Note that if a graph G is k-connected, then G is a k-core but the reverse is not true. The following lemma directly follows from the definition of k-d-CDS and will be used extensively to prove properties of k-d-CDS.

Lemma 1. *Consider a connected graph $G = (V, E)$ with two integers $d \geq k \geq 1$ and a vertex-cut $C \subset V$ of G. Then $|S \cap C| \geq k$ for every k-d-CDS $S \subset V$ of G.*

Proof. Suppose there is a k-d-CDS $S \subset V$ such that $|S \cap C| \leq k-1$. Then, obviously $G[S]$ is not k-connected since removing all vertices in $S \cap C$ from S would render $G[S]$ disconnected. $\square$

Theorem 7. *Consider a connected graph $G = (V, E)$ with two integers $d \geq k \geq 1$. A set $S \subset V$ is a k-d-CDS of G if and only if*

1. $|S \cap C| \geq k$ *for every vertex-cut* $C \subset V$*, and*

2. $|S \cap N(v)| \geq d \; \forall v \in V \setminus S$.

Proof. The sufficient condition follows directly from the definition of k-d-CDS and Lemma 1. For necessary condition, if $|S \cap C| \geq k$ for every vertex-cut $C \subset V$, at least k vertices must be removed to make $G[S]$ disconnected, meaning that $G[S]$ is k-connected. If $|S \cap N(v)| \geq d$ for every vertex $v \in V \setminus S$, then obviously S is d-dominating. Thus, S is a k-d-CDS for G. □

As a special case of Theorem 7, when $k = d$, a set $S \subset V$ is a k-k-CDS for G if and only if $|S \cap C| \geq k$ for every vertex-cut $C \subset V$. It is easy to see that a graph G has a CDS if and only if G is connected. The following theorem generalizes this property.

Theorem 8. *Given a graph $G = (V, E)$ and two integers $d \geq k \geq 1$, the following are equivalent.*

1. *There is a k-d-CDS in G.*

2. *The vertex set V is a k-d-CDS for G.*

3. *The graph G is k-connected.*

Proof. 2 and 3 are equivalent from Theorem 7. It is easy to see that 2 implies 1. We prove that 1 implies 2. Suppose that there is a k-d-CDS $S \subset V$ for G. From Theorem 7, this implies that $|S \cap C| \geq k$ for every vertex-cut $C \subset V$ and $|S \cap N(v)| \geq d$ for every vertex $v \in V \setminus S$. Then, obviously, V is also a k-d-CDS for

G, since $|V \cap C| \geq |S \cap C| \geq k$. Note that $V \setminus V$ is empty and clearly d-domination is satisfied for every vertex in $V \setminus V$. □

For a vertex-cut $|C| \leq n - 2$, the following lemma holds and this will be used extensively to prove several propositions.

Lemma 2. *[71] Given a graph $G = (V, E)$, a vertex-cut $C \subset V$ is minimal if and only if every vertex in C has a neighbor in every connected component of $G[V \setminus C]$.*

Lemma 3. *For an integer $k \geq 1$, suppose a graph $G = (V, E)$ has an $(k+1)$-$(k+1)$-CDS $S \subset V$. Then for any $v \in S$, $S \setminus \{v\}$ is an k-k-CDS for G.*

Proof. Let $S \subset V$ be a minimum $(k+1)$-$(k+1)$-CDS. We show that, for any $v \in S$, $S \setminus \{v\}$ is a k-k-CDS. Since S is $(k+1)$-connected, removal of any vertex $v \in S$ implies that $S \setminus \{v\}$ is k-connected. Any vertex in $V \setminus S$ is dominated by at least k vertices in $S \setminus \{v\}$ since it was originally dominated by $k+1$ vertices. Also, the vertex v is dominated by at least k vertices in $S \setminus \{v\}$. Thus, $S \setminus \{v\}$ is a k-k-CDS. □

Robustness is a desirable property. However, the following Corollary implies that the robustness is not free. In fact, enforcing robustness accompanies a strictly more-costly solution.

Corollary 1. *For an integer $k \geq 1$, suppose that a graph G has an $(k+1)$-$(k+1)$-CDS. Let $\gamma_{k,d}(G)$ and $\gamma_c(G)$ be the size of a Mk-d-CDS and the size of a MCDS of G, respectively. Then,*

$$k + \gamma_c(G) \leq 1 + \gamma_{k,k}(G) \leq \gamma_{k+1,k+1}(G).$$

Proof. The rightmost inequality holds directly from Lemma 3. To show the first inequality, we use induction. For $k = 1$, we have $1 + \gamma_c(G) \leq 1 + \gamma_{1,1}(G)$ since

$\gamma_c(G) = \gamma_{1,1}(G)$. Suppose $k + \gamma_c(G) \leq 1 + \gamma_{k,k}(G)$ holds for k and show that the inequality holds for $k+1$. For $k+1$, we have $k + 1 + \gamma_c(G) \leq 1 + \gamma_{k+1,k+1}(G)$ or $k + \gamma_c(G) \leq 1 + \gamma_{k,k}(G)$. □

Note that each inequality in Corollary 1 is sharp on graphs which is obtained by deleting a perfect matching from K_{2k+4}. Figure 4.3 illustrates this for $k = 2$.

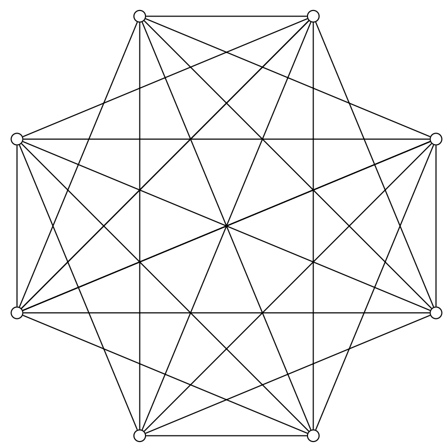


Figure 4.3: Illustration of sharpness of inequality in Lemma 1 for $k = 2$.

From Lemma 2, we can easily see that for a given connected graph G and a minimal cutset $C \subset V$, $(V \setminus C) \cup \{v\}$ is a 1-1-CDS $\forall v \in C$. This statement holds for any graph G with connectivity at least 1. However, we cannot generalize this to k-k-CDS for $k \geq 2$. In other words, given a graph G with connectivity at least k and a minimal vertex-cut $C \subset V$, $(V \setminus C) \cup C'$ is not necessarily a k-k-CDS, where $C' \subset C$ with $|C'| = k$. In Figure 4.4, the graph G is triconnected and $C = \{3, 5, 7, 9\}$ is a minimal vertex-cut of G. However, the vertex set $(V \setminus C) \cup \{7, 9\}$ is not 2-2-CDS since removal of either vertex 8 or 10 would render $G[(V \setminus C) \cup \{7, 9\}]$ disconnected.

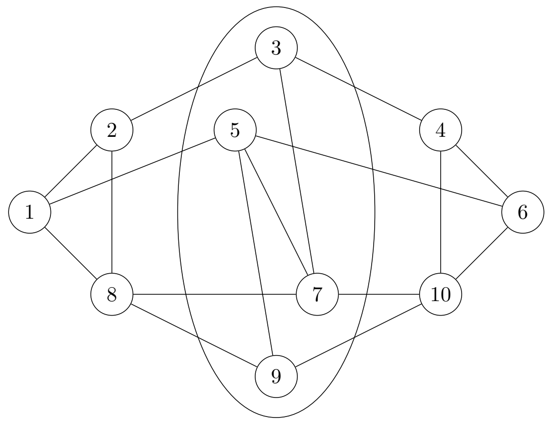


Figure 4.4: Fixing two vertices from a minimal vertex-cut does not necessarily give a 2-2-CDS.

4.3 IP formulation

Depending on the values of k and d, we present three different formulations. In each formulation, x is the characteristic vector of a k-d-CDS of a graph $G = (V, E)$. For $d = k \geq 1$, we have the following formulation for Mk-d-CDS problem.

$$\gamma_{k,k}(G) = \min \sum_{v \in V} x_v \tag{4.1}$$

$$\sum_{v \in C} x_v \geq k \text{ for every minimal vertex-cut } C \subset V \tag{4.2}$$

$$x_v \in \{0, 1\} \ \forall v \in V \tag{4.3}$$

This is the generalization of the formulation for MCDS problem presented in [105]. The formulation is still valid when we replace the constraints (4.2) for every vertex-cut. However, the minimal vertex-cuts subsumes the non-minimal vertex-cut. Notice that the above formulation could have exponentially many constraints since there exist k-connected graphs with $\Omega(2^k \frac{n^2}{k^2})$ minimum vertex-cuts [64], giving a lower bound for the number of minimal vertex-cuts.

For $d > k \geq 1$, we need to add some other constraints to the above formulation to guarantee d-domination for vertices not in the k-d-CDS.

$$\gamma_{k,d}(G) = \min \sum_{v \in V} x_v \tag{4.4}$$

$$\sum_{v \in C} x_v \geq k \text{ for every minimal vertex-cut } C \subset V \tag{4.5}$$

$$dx_v + \sum_{j \in N(v)} x_j \geq d \;\forall v \in V \tag{4.6}$$

$$x_v \in \{0,1\} \;\forall v \in V \tag{4.7}$$

For $k > d \geq 1$, the vertex-cut constraints that we used for the previous two formulations does not work. Instead, we need to modify the constraints as follows.

$$\gamma_{k,d}(G) = \min \sum_{v \in V} x_v \tag{4.8}$$

$$\sum_{v \in S} x_v \geq kx_a x_b \text{ for every minimal } a\text{-}b\text{-separator } S \subset V \tag{4.9}$$

$$dx_v + \sum_{j \in N(v)} x_j \geq d \;\forall v \in V \tag{4.10}$$

$$x_v \in \{0,1\} \;\forall v \in V \tag{4.11}$$

In constraint (4.9), for two vertices $a, b \in V$, an a-b-separator is a subset of vertices such that the removal renders the graph disconnected; also a and b belong to different connected components. If needed, the constraints (4.9) can be linearized.

4.4 k-d-CDS polytope

From Lemma 1 we know that if $C \subset V$ is a vertex-cut of a graph G, then $|S \cap C| \geq k$ for every k-d-CDS $S \subset V$ in G. Validity of vertex-cut inequality derives directly from this observation. Note that we can generalize vertex-cut inequality in the following way.

Property 2. *(Generalized cutset inequalities). Consider a connected graph $G = (V, E)$ with two integers $d \geq k \geq 1$ and a subset $C \subset V$. Then the following inequality is valid for k-d-CDS polytope.*

$$\sum_{i \in C} x_i \geq k - \kappa(G[V \setminus C]).$$

Proof. We prove this using contradiction. Suppose that there is a k-d-CDS $S \subset V$ such that $|S \cap C| < k - \kappa(G[V \setminus C])$. Let $C' = C \setminus S$ be the set of vertices in C that are not in S. Then, $\kappa(G[V \setminus C']) = \kappa(G[(V \setminus C) \cup (S \cap C)]) \leq \kappa(G[V \setminus C]) + |S \cap C| < k$. Note that $S \subset V \setminus C'$. From Theorem 8, there does not exist a k-d-CDS in $G[V \setminus C']$, a contradiction. □

CDS polytope is known to be full dimensional if and only if the graph is biconnected. We can generalize this result for k-d-CDS polytope. Theorem 9 and 10 are stated without proof since they are rather straightforward.

Theorem 9. *Given a graph $G = (V, E)$ with two integers $d \geq k \geq 1$, k-d-CDS polytope is full dimensional if and only if G is $(k+1)$-connected and $\delta(G) \geq d$.*

Theorem 10. *Given a graph $G = (V, E)$ with two integers $d \geq k \geq 1$, if G is $(k+1)$-connected and $\delta(G) \geq d$, then the following holds:*

1. *$x_v \leq 1$ induces a facet.*

2. *$x_v \geq 0$ induces a facet if and only if*

   i. $v \notin C$ for any vertex-cut C with $|C| = k + 1$, *ii.* $|N(i)| \geq d + 1 \ \forall i \in N[v]$.

In general, for a k-connected graph $G = (V, E)$ with $|V| = n$ and two integers $d \geq k \geq 1$, k-d-CDS polytope P has dimension $n - |\Theta_{k,d}(G)|$, where $\Theta_{k,d}(G) = \{v \in V \mid \text{v belongs to a vertex-cut of size} k \text{ or } deg(v) < d\}$.

Notice that the separation problem for the constraints (4.2) and (4.5) is a weighted vertex-connectivity problem [55]. Suppose that we have current solution $\bar{x} = [0, 1]^n$ for k-d-CDS. We construct an instance of the weighted vertex connectivity problem by assigning $\bar{x}_w$ for each vertex $w \in V$. It is easy to see that the weighted vertex connectivity of G is at least k if and only if all the vertex-cut inequalities are satisfied. If the weighted vertex connectivity is less than k, we can find the vertex-cut C with weights less than k in $O(\kappa nm \log(\frac{n^2}{m}))$ as shown in [55]. The corresponding inequality $\sum_{i \in C} x_i \geq k$ is valid and will separate the current solution $\bar{x}$ from the k-d-CDS polytope. Thus, for $d \geq k$, we can solve the linear programming relaxation of Mk-d-CDS problem in polynomial time (e.g. ellipsoid method).

Minimal vertex-cut inequalities are not necessarily facet inducing. For instance, take the 1-1-CDS polytope of a cycle of 6 vertices (C_6) shown in Figure 4.5. The minimal vertex-cut inequality $x_{v_1} + x_{v_3} \geq 1$ is subsumed by the inequality $x_{v_1} + x_{v_3} + x_{x_5} \geq 2$, which induces a facet and is called 2-robust vertex-cut according to the following definition. Motivated by the fact that minimal vertex-cut may not induce facet, we define the following generalization of vertex-cut called r-robust vertex-cuts.

Definition 11. *A set $C \subset V$ is called r-robust vertex-cut if $C \setminus C'$ is vertex-cut for any $C' \subset C$ with $|C'| < r$, where r is a positive integer.*

Proposition 4. *Let $d \geq k \geq 1$ be positive integers. Given a r-robust vertex-cut*

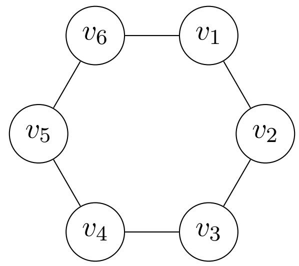


Figure 4.5: Cycle of 6 vertices.

$C \subset V$, the following inequality is valid for k-d-CDS.

$$\sum_{i \in C} x_i \geq k + r - 1$$

Proof. We prove this by induction on r. When $r = 1$, this inequality is the vertex-cut inequality. Suppose that the r-robust inequality holds and we show that the $(r+1)$-robust inequality holds. Let $C \subset V$ be an r-robust vertex-cut. Then $C \setminus \{v\}$ is r-robust vertex-cut for any $v \in C$ and we have the inequality $\sum_{i \in C \setminus \{v\}} x_i \geq k + r - 1$ for any vertex $v \in C$. Summing these inequalities gives

$$\sum_{i \in C} (|C| - 1) x_i \geq |C|(k + r - 1),$$

and dividing both sides by $|C| - 1$ gives the following.

$$\sum_{i \in C} x_i \geq \left\lceil \frac{|C|}{|C| - 1}(k + r - 1) \right\rceil = \left\lceil \frac{|C| - 1}{|C| - 1}(k + r - 1) + \frac{1}{|C| - 1}(k + r - 1) \right\rceil = k + r.$$

□

As we define minimal vertex-cut, we can also define minimal r-robust vertex-cut. An r-robust vertex-cut $C \subset V$ is minimal if no proper subset of C is an r-robust vertex-cut. (i.e., an r-robust vertex-cut C is minimal if $C \setminus \{v\}$ is not r-robust vertex-cut for every $v \in C$). From Corollary 2, we know that if C is minimal vertex-cut for a connected graph G, then $(V \setminus C) \cup \{v\}$ $\forall v \in C$ is CDS. From this, one might expect that if C is a minimal r-robust vertex-cut, then $(V \setminus C) \cup C'$ would give a CDS for any $C' \subset C$ with $|C'| = r$. However, this does not necessarily hold. For instance, in Figure 4.6, $C = \{4, 5, 6, 7, 8, 9\}$ is a minimal 2-robust vertex-cut. However, $(V \setminus C) \cup \{u, v\}$ is not a CDS, where two vertices $u, v \in C$ are not adjacent.

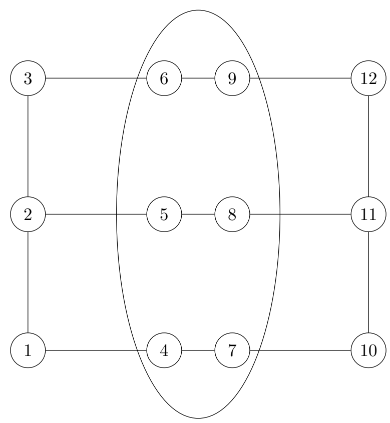


Figure 4.6: Given a connected graph G and a minimal vertex-cut $C \subset V$, fixing two vertices from C may not give a CDS.

Proposition 5. *Given a biconnected graph $G = (V, E)$ with $\gamma(G) \geq 2$ and a vertex-cut $C \subset V$, the inequality $\sum_{i \in C} x_i \geq 1$ induces a facet for* 1-1-*CDS polytope of G if and only if*

1. *C is minimal, and*

2. *$C \cup \{v\}$ is not a* 2*-robust vertex-cut for every $v \in V \setminus C$.*

Proof. From Theorem 9, we know that if G is biconnected, then 1-1-CDS polytope is full dimensional. Let P be the 1-1-CDS polytope of G and $F = \{x \in P | \sum_{i \in C} x_i = 1\}$. We provide n affinely independent points in F. We generate $|C|$ points first by considering $S_v = (V \setminus C) \cup \{v\}$ for each vertex $v \in C$. We claim S_v is 1-1-CDS for every $v \in C$. S_v is 1-connected since v has a neighbor in every connected component of $G[V \setminus C]$ by minimality of C and Lemma 2. S_v is dominating since all vertex not in S_v is in C, and by minimality of C, every vertex in C has a neighbor in every connected component of $G[V \setminus C]$.

Next, we generate $n - |C|$ points in F. Condition 2 says that for every vertex $v \in V \setminus C$, $C \cup \{v\}$ is not a 2-robust vertex-cut. This implies that for every $v \in V \setminus C$, there is $w \in C$ such that $(C \cup \{v\}) \setminus \{w\}$ is not a vertex-cut. Let $S_v = V \setminus ((C \cup \{v\}) \setminus \{w\})$. Then we claim S_v is 1-1-CDS for G. As stated above, we know that $(C \cup \{v\}) \setminus \{w\}$ is not a vertex-cut and $G[S_v]$ is connected. Now we claim that S_v is dominating. Every vertex $z \in C$ is dominated since it has a neighbor in every connected component of $G[V \setminus C]$. Note that there are at least two such components and z remains dominated even if v is one of its neighbors. Every vertex in $V \setminus C$ is obviously dominated since it belongs to S_v. Finally, we show that v is dominated by S_v. If v is isolated in $G[V \setminus C]$, then v is adjacent to every vertex in C, implying that it is dominated since G is biconnected. If it is not isolated in $G[V \setminus C]$, then it has a neighbor in its connected component which belongs to S_v. We can easily show that these n points are affinely independent by showing that they are linearly independent.

To prove the other direction, we show that if condition 1 or 2 fails, then the vertex-cut inequality $\sum_{i \in C} x_i \geq 1$ cannot induce facet when $dim(P) = n$. Suppose that C is not minimal, meaning that there is a vertex $v \in C$ such that $C \setminus \{v\}$ is a vertex-cut. Then the inequality $\sum_{i \in C \setminus \{v\}} x_i \geq 1$ subsumes the inequality $\sum_{i \in C} x_i \geq 1$. For the other case, suppose C is minimal, but there exist $v \in V \setminus C$ such that $C \cup \{v\}$ is a 2-robust vertex-cut. Then, the 2-robust vertex-cut inequality $x_v + \sum_{i \in C} x_i \geq 2$ subsumes the inequality $\sum_{i \in C} x_i \geq 1$. $\square$

4.5 Numerical experiments

In this section, we describe the procedure for solving the Mk-k-CDS problem and show the computational results for $k = 1, 2, 3$, and 4. As stated in section 4.3, the mathematical programming formulation for Mk-k-CDS problem has exponentially many constraints. Thus, we do not include all of the constraints a priori. Instead, we start with the k-total-domination constraints and add the constraints ensuring k-connectivity in a lazy fashion.

- **Initial constraints**: *k-total-domination.* For each vertex $i \in V$ add the k-total-domination constraint $\sum_{j \in N(i)} x_j \geq k$. This is valid since $N(i)$ is a vertex-cut for every vertex $i \in V$.

- **Lazy constraints**: *k-connectivity.* Whenever a feasible solution (binary solution), say $\bar{x}$, which satisfies all the constraints added so far is found, connectivity is checked. Let $S = \{i \in V | \bar{x}_i = 1\}$ and $\bar{S} = V \setminus S$. If $G[S]$ is not connected, then $\bar{S}$ is a vertex-cut for G and $\sum_{i \in \bar{S}} x_i \geq k$ is valid and can be added. However, this inequality could be very weak, and we can strengthen it by finding a minimal subset of $\bar{S}$ which is a vertex-cut for G. For detailed procedure for making a vertex-cut minimal, refer to Remark 3. If $G[S]$ is connected, but not k-connected, then we find a minimal vertex-cut $C \subset S$ of $G[S]$. Note that $C \cup \bar{S}$ is a vertex-cut for G. We can strengthen it as stated in Remark 3.

We illustrate the solution procedure for minimum 3-3-CDS problem in detail for a triconnected graph G as shown in Figure 4.7. Initially, we solve the following minimum 3-total dominating set problem.

$$\min \sum_{i \in V} x_i \tag{4.12}$$

$$\sum_{j \in N(i)} x_j \geq 3 \ \forall i \in V \tag{4.13}$$

$$x_i \in \{0, 1\} \ \forall i \in V \tag{4.14}$$

Suppose an optimal solution for this problem is $S = \{1, 2, 3, 5, 7, 8, 9\}$. In Figure 4.8, vertices in S is shown in bold. We check if $G[S]$ is triconnected. Clearly $G[S]$ is not triconnected and $C = \{9\}$ is a vertex-cut of $G[S]$. Let $\bar{S} = V \setminus S = \{4, 6, 10, 11\}$. Then $C \cup \bar{S} = \{4, 6, 9, 10, 11\}$ is a vertex-cut for G. Now we add the following valid inequality.

$$\sum_{i \in \bar{S}} x_i \geq 3$$

Suppose that an optimal solution that satisfies all constraints added so far is $S_1 = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$ as shown in Figure 4.9. Let $\bar{S}_1 = \{10, 11\}$. Note that $G[S_1]$ is not triconnected and $C = \{9\}$ is a vertex-cut for $G[S_1]$. Now we have a vertex-cut $C \cup \bar{S}_1 = \{9, 10, 11\}$ for G. We add the following vertex-cut inequality.

$$\sum_{i \in C \cup \bar{S}_1} x_i \geq 3$$

Suppose that an optimal solution that satisfies all constraints added so far is $S_2 = \{1, 2, 3, 5, 7, 8, 9, 10, 11\}$ as shown in Figure 4.10. Note that this is triconnected and is an optimal solution to minimum 3-3-CDS problem.

Remark 3. *(Making a vertex-cut minimal in $O(n^2)$ time.) Given a vertex-cut $C \subset V$ and a graph $G = (V, E)$, let $C' \subset C$ be the set of vertices which are adjacent to a*

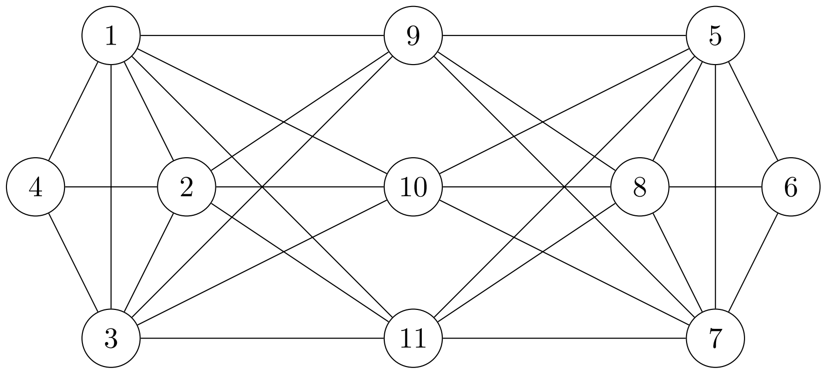


Figure 4.7: A graph G with $\kappa(G) = 3$.

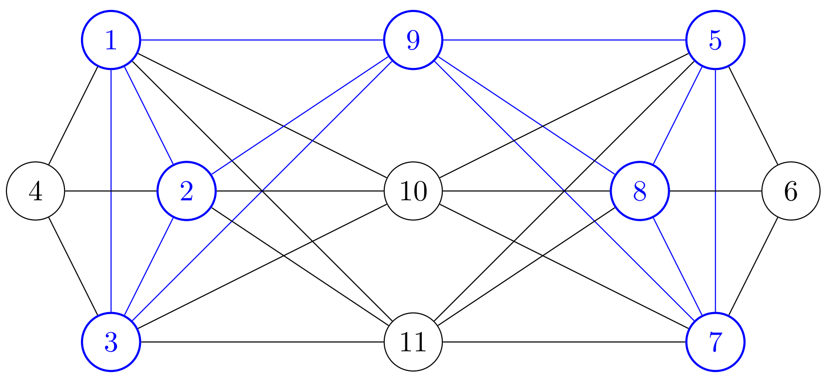


Figure 4.8: $S = \{1, 2, 3, 5, 7, 8, 9\}$ is a 3-total dominating set of G while $G[S]$ is not triconnected.

vertex in $V \setminus C$. Let $\boldsymbol{S} := \{S_1, \ldots, S_p\}$ be the collection of connected components of $G[V \setminus C']$, where $S_i \subset V \setminus C'$, $i = 1, \ldots, p$. For each vertex $v \in C'$, perform the following. If v has a neighbor in every connected component, then do nothing. Otherwise, merge v and all components from $\boldsymbol{S}$ that v has a neighbor in and update $C' = C' \setminus \{v\}$. The time complexity of this procedure is $O(n^2)$. When the procedure is completed, it is clear that every vertex $v \in C'$ would have a neighbor in every connected component of $G[V \setminus C']$. Obviously, from Lemma 2, C' is minimal.

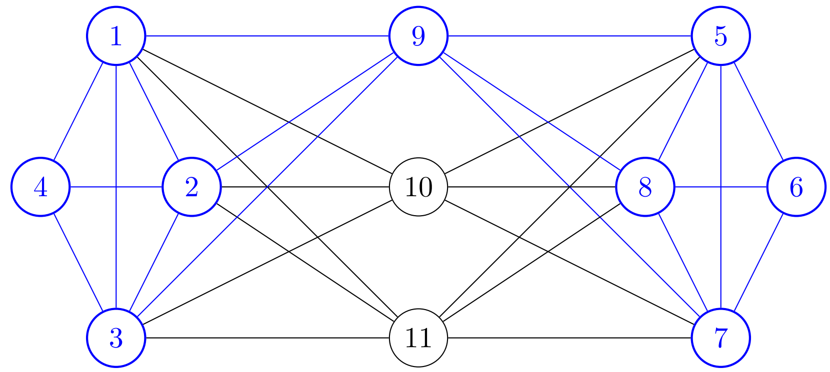


Figure 4.9: $S = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$ is a 3-total dominating set of G while $G[S]$ is not triconnected.

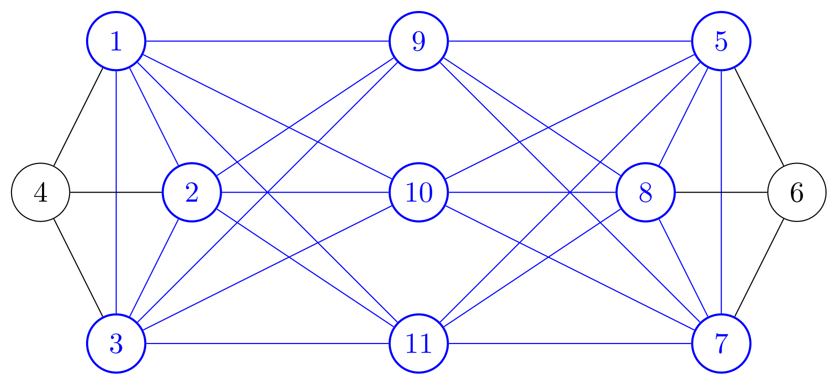


Figure 4.10: $S = \{1, 2, 3, 5, 7, 8, 9, 10, 11\}$ is triconnected and it is an optimal 3-3-CDS.

Given a vertex-cut $\bar{S} = V \setminus S = \{4, 6, 9, 10, 11\}$ in Figure 4.8, we can use the procedure in Remark 3 to have a minimal vertex-cut $\{9, 10, 11\}$. This would have allowed us to add a stronger valid inequality and skip the procedure shown in Figure 4.9.

4.5.1 Computational setup and numerical test results

All numerical experiments are performed on a *Dell Precision WorkStation T7500*® computer with eight 2.40 GHz Intel Xeon® processors and 12 GB RAM. Gurobi

Optimizer® version 5.5 is used with its lazy-constraint callback [59].

Fand and Watson [34] provide several mathematical programming formulations for MCDS problem with computational experiment results. The fastest one utilizes Miller-Tucker-Zemlin constraints to impose the connectivity. Table 4.1 presents the time comparisons between the fastest formulation from [34] and our approach to solve MCDS problem with optimal objective function value.

Table 4.1: Running time comparison with Fan and Watson (FW) [34] for solving the minimum 1-1-CDS problem.

Graph	opt	FW time	Our time
IEEE-14	5	0.02	0.01
IEEE-30	11	0.22	0.01
IEEE-57	31	200.59	1.07
IEEE-118	43	699.83	0.08
IEEE-300	129	5033.97	52.88
RTS-96	32	445.69	0.69

Simonetti, Cunha, and Lucena (SCL) [94] and Lucena, Maculan, and Simonetti (LMS) [78] studies exact approaches for MCDS problem and present the numerical experiment results on the same instances. In Table 4.2, we compare the running time of SCL and LMS to our approach. A dash in Table 4.2 indicates unsolved in time limit. SCL and LMS have time limit of 3,600 seconds and 18,000 seconds, respectively. The graph v50_d20 has 50 nodes and 20% density.

In Table 4.3, we present the optimal solution and the running time for both minimum k-total dominating set problem and the Mk-k-CDS problem for $k = 1, 2, 3,$ and 4. Note that blank entries denote that the graph has no solution. As we can see from Table 4.3, for most of the considered instances, the time taken to solve

Table 4.2: Running time comparison with Simonetti, Cunha, and Lucena (SCL) [94] and Lucena, Maculan, and Simonetti (LMS) [78] for solving minimum 1-1-CDS problem.

Graph	opt	SCL time	LMS time	Our time
v30_d10	15	0.01	0.01	0.24
v30_d20	7	0.02	0.10	0.01
v30_d30	4	0.05	0.03	0.01
v30_d50	3	0.04	0.08	0.01
v30_d70	2	0.02	0.01	0.01
v50_d5	31	0.02	0.01	0.59
v50_d10	12	0.42	0.36	0.12
v50_d20	7	0.66	1.32	0.08
v50_d30	5	0.25	1.21	0.07
v50_d50	3	0.25	0.51	0.01
v50_d70	2	0.29	0.04	0.02
v70_d5	27	1.42	0.26	1.41
v70_d10	13	34.29	4.73	0.09
v70_d20	7	2.16	16.30	0.15
v70_d30	5	1.00	2.90	0.17
v70_d50	3	0.70	1.33	0.01
v70_d70	2	0.79	1.92	0.07
v100_d5	24	342.25	12.50	0.36
v100_d10	13	32.11	9.36	0.34
v100_d20	8	174.93	86.16	0.40
v100_d30	6	193.65	258.15	0.94
v100_d50	4	35.41	132.55	0.70
v100_d70	3	12.03	154.10	1.27
v120_d5	25	-	2.65	0.31
v120_d10	13	-	65.49	0.34
v120_d20	8	610.89	393.47	1.86
v120_d30	6	475.54	653.70	2.32
v120_d50	4	168.55	815.64	1.64
v120_d70	3	31.67	356.31	2.44
v150_d5	26	-	2954.00	3.46
v150_d10	14	-	3247.89	4.72
v150_d20	9	-	-	9.34
v150_d30	6	1954.00	2317.35	6.54
v150_d50	4	481.61	2756.36	2.41
v150_d70	3	43.75	1828.86	4.77
v200_d5	27	-	-	32.92
v200_d10	16	-	-	496.43
v200_d20	9	-	-	243.25
v200_d30	7	-	-	172.55
v200_d50	4	2249.43	20155.00	8.16
v200_d70	3	271.90	8154.13	9.45

minimum k-total dominating set problem and Mk-k-CDS problem are comparable. This implies that the connectivity constraints may not be burdensome.

Table 4.3: A comparison of running times for minimum k-total dominating set and Mk-k-CDS problem.

	1TDS		1-1-CDS		2TDS		2-2-CDS		3TDS		3-3-CDS		4TDS		4-4-CDS	
Graph	opt	time	opt	time	opt	time	opt	time	opt	time	opt	time	opt	time	opt	time
v30_d10	12	0.01	15	0.24												
v30_d20	6	0.01	7	0.01												
v30_d30	4	0.01	4	0.01	8	0.05	8	0.06	11	0.00	11	0.05	15	0.02	15	0.02
v30_d50	3	0.01	3	0.01	5	0.02	5	0.00	7	0.02	7	0.00	9	0.02	9	0.02
v30_d70	2	0.01	2	0.01	4	0.03	4	0.03	5	0.02	5	0.02	6	0.06	6	0.05
v50_d5	21	0.01	31	0.59												
v50_d10	11	0.01	12	0.12	22	0.00	22	0.00								
v50_d20	7	0.10	7	0.08	12	0.08	12	0.08	17	0.02	17	0.02				
v50_d30	5	0.18	5	0.07	8	0.20	8	0.06	12	0.08	12	0.28	15	0.03	15	0.09
v50_d50	3	0.01	3	0.01	5	0.06	5	0.05	7	0.02	7	0.05	9	0.02	9	0.02
v50_d70	2	0.01	2	0.02	4	0.05	4	0.09	5	0.02	5	0.05	7	0.14	7	0.11
v70_d5	23	0.01	27	1.41	47	0.00	47	0.00								
v70_d10	13	0.18	13	0.09	23	0.02	24	0.09								
v70_d20	7	0.13	7	0.15	12	0.13	12	0.14	17	0.23	17	0.20	23	0.13	23	0.13
v70_d30	5	0.15	5	0.17	8	0.19	8	0.20	12	0.19	12	0.19	15	0.14	15	0.14
v70_d50	3	0.02	3	0.01	5	0.23	5	0.20	7	0.19	7	0.13	9	0.23	9	0.20
v70_d70	2	0.07	2	0.07	4	0.36	4	0.36	5	0.17	5	0.50	7	0.27	7	0.39
v100_d5	23	0.11	24	0.36	44	0.08	44	0.08								
v100_d10	13	0.26	13	0.34	24	0.27	24	0.48	35	0.25	35	0.27				
v100_d20	8	0.47	8	0.40	13	0.45	13	0.45	18	0.27	18	0.34	23	0.48	23	0.50
v100_d30	6	0.72	6	0.94	9	0.78	9	1.89	13	2.34	13	2.48	16	1.12	16	1.22
v100_d50	4	0.67	4	0.70	6	0.50	6	0.62	8	0.69	8	0.81	10	0.58	10	0.64
v100_d70	3	1.07	3	1.27	4	0.55	4	0.58	6	0.58	6	0.67	7	0.97	7	0.69
v120_d5	25	0.21	25	0.31	46	0.16	46	0.16								
v120_d10	13	0.41	13	0.34	24	1.11	24	0.98	35	1.84	35	1.97	46	0.69	46	0.94
v120_d20	8	2.63	8	1.86	13	2.04	13	2.48	18	1.09	18	6.72	23	1.40	23	1.19
v120_d30	6	2.43	6	2.32	9	3.01	9	3.03	12	5.16	12	2.84	16	5.20	16	6.38
v120_d50	4	1.30	4	1.64	6	1.76	6	1.81	8	2.17	8	2.31	10	1.89	10	1.95
v120_d70	3	2.12	3	2.44	4	1.62	4	1.94	6	1.34	6	2.76	7	0.97	7	0.92
v150_d5	24	0.49	26	3.46	45	0.31	45	0.34	71	0.28	71	0.28				
v150_d10	14	2.91	14	4.72	24	2.92	24	6.19	35	5.40	35	6.71	46	31.01	46	31.12
v150_d20	8	2.03	9	9.34	14	27.55	14	28.11	19	79.65	19	80.03	24	123.77	24	121.24
v150_d30	6	5.31	6	6.54	10	17.29	10	14.10	13	136.02	13	56.01	17	320.29	17	329.66
v150_d50	4	2.46	4	2.41	6	3.11	6	3.23	8	5.04	8	2.40	10	3.35	10	3.32
v150_d70	3	3.74	3	4.77	4	4.65	4	4.93	6	2.54	6	3.20	7	2.29	7	2.31
v200_d5	27	11.49	27	32.92	48	10.41	48	9.36	71	4.35	71	5.24				
v200_d10	16	775.14	16	496.43	26	2018.38	26	2111.52	36	3026.18	36	3103.52	46	2846.70	46	1802.14
v200_d20	9	230.20	9	243.25	14	739.68	14	832.97	19	1383.64	19	21766.20	24	4074.83	24	4171.82
v200_d30	6	162.12	7	172.55	10	227.16	10	315.61	14	5280.93	14	5623.75	17	3936.37	17	4040.89
v200_d50	4	8.41	4	8.16	6	40.14	6	33.34	8	418.34	8	246.05	11	1008.54	11	1063.84
v200_d70	3	9.31	3	9.45	5	44.41	5	35.63	6	8.28	6	9.08	8	136.03	8	132.24
mean		29.92		24.63		82.89		89.77		314.16		937.12		430.95		403.94

5. SELECTIVE DOMINATING SET

This section introduces several variations of the classical graph-theoretic concept of domination that are motivated by practical considerations. Computational complexity of the decision versions of the corresponding optimization problems is analyzed showing that these variations are hard in their own respect. We also establish some basic properties of the corresponding polyhedra and develop analytical bounds on the size of structures of interest. Numerical experiment results using ILOG CPLEX 12.1® on random unit disk graphs indicate that some variations are much more challenging to solve than others. We also compare the performance of simulated annealing against CPLEX 12.1®. For almost all considered instances, simulated annealing outperforms CPLEX 12.1®.

5.1 Variations of dominating set

One of the most popular applications of dominating set arises in wireless sensor placement problem, where each vertex of a graph $G = (V, E)$ represents a site that has to be monitored, and two vertices are connected by an edge if both of the corresponding sites can be monitored by a single sensor placed in one of the two sites. In order to monitor all sites of interest with a minimum number of sensors, one needs to find a minimum dominating set in the graph G. In some realistic scenarios, however, there could be a situation where a sensor cannot be placed because of numerous reasons. Or, different sites may require a greater number of neighboring sensors if a sensor is not placed on it. With these cases in mind, we introduce some varieties of dominating set called *selective dominating set*, *generalized selective dominating set*, *mixed selective dominating set*, and *generalized dominating set*.

Selective dominating set Consider a sensor placement problem with the following considerations: (a) there may be sites where locating a sensor is physically impossible; and (b) not all the sites in the network may need to be monitored, while they can still be used for locating sensors. To address these issues, we define the concept of a *selective dominating set* as follows.

Definition 12. *For a graph $G = (V, E)$ and a vector $r \in \{-1, 0, 1\}^n$ a subset of vertices D is called a selective dominating set corresponding to r, or an r-SDS, if (a) $i \notin D$ whenever $r_i = -1$; and (b) any $i \in V - D$ must have a neighbor in D, unless $r_i = 0$. The minimum selective dominating set (MSDS) problem is to find an r-SDS of minimum size in G.*

We call the vector r the *domination requirement*. The domination requirement of the classical minimum domination set problem is vector of 1's. Note that this definition implies that a vertex $i \in V$ with $r_i = -1$ must be dominated by at least one vertex from r-SDS, i.e., we do not consider the "useless" vertices that cannot serve as sensor locations and, at the same time, do not need to be monitored. Given the domination requirement vector r, we can easily check if the MSDS problem has a feasible solution or not. Throughout this section, we assume that G a has feasible r-SDS.

Generalized selective dominating set Suppose that some of the monitored sites may be more important than others and, if not used to place a sensor, may require locating of sensors in multiple neighboring sites. This motivates a further generalization of the notion of SDS by defining a generalized selective dominating set as follows.

Definition 13. *For a graph $G = (V, E)$ and a vector $r \in [-\Delta(G), \Delta(G)]^n \cap \mathbb{Z}^n$, a subset of vertices D is called a generalized selective dominating set corresponding to r, or an r-GSDS, if (a) $i \notin D$ whenever $r_i < 0$; and (b) any $i \in V - D$ must* *have at least r_i neighbors in D, unless $r_i = 0$. The minimum generalized selective dominating set (MGSDS) problem is to find an r-GSDS of minimum size in G.*

For a vertex $i \in V$, let $d_r^+(i) = |\{j \in N(i) : r_j \geq 0\}|$ denote the number of neighbors of i that can be included in an r-GSDS of G. Note that if $r_i < -d_r^+(i)$ then there is no feasible r-GSDS in G. Also, $r_i > d_r^+(i)$ implies that $i \in D$ for any r-GSDS D, in which case we can remove i from G and r_i from r, reduce the values of all r_j, such that $(i, j) \in E$, by one, and obtain a smaller equivalent problem instance. Hence, it is reasonable to consider only domination requirement vectors r from the set $\mathcal{R}_G := \{r \in \mathbb{Z}^n : -d_r^+(i) \leq r_i \leq d_r^+(i), i = 1, \ldots, n\}$. Throughout this paper, we assume G has a feasible r-GSDS.

Mixed selective dominating set There could be a site which is extremely important so that not only must a sensor be placed on it, but also at least one additional sensor on one of its neighboring sites is needed. This motivates another generalization of SDS by defining a mixed selective dominating set as follows.

Definition 14. *For a graph $G = (V, E)$ and a vector $r \in \{-1, 0, 1, +1\}^n$ a subset of vertices D is called a mixed selective dominating set corresponding to r, or an r-MSDS, if (a) $i \notin D$ whenever $r_i = -1$; and (b) $i \in D$ and i must have a neighbor in D whenever $r_i = +1$; and (c) any $i \in V - D$ must have a neighbor in D, unless $r_i = 0$. The minimum mixed selective dominating set (MMSDS) problem is to find an r-MSDS of minimum size in G.*

Here $+1$ is used to emphasize that if $r_i = +1$, then both i and at least one vertex from $N(i)$ must be included in r-MSDS. This requirement can be considered as a greater degree of domination. Given a graph $G = (V, E)$ and $r \in \{-1, 0, 1, +1\}^n$, the MMSDS problem can be infeasible. For example, for a vertex i with $r_i = -1$ if every $j \in N(i)$ has $r_j = -1$ then the domination requirement of i cannot be satisfied and the problem becomes infeasible. Or, for a vertex i with $r_i = +1$ if every $j \in N(i)$ has $r_i = -1$, then the i's domination requirement cannot be satisfied and the problem becomes infeasible. Feasibility can be checked easily, and throughout this chapter we assume G has feasible r-MSDS.

Generalized dominating set We consider a situation where $1 \leq r_i \leq deg_G(i)$ for $i \in V$. This is another way of imposing greater degree of domination and is introduced by Harant [49] in the following way.

Definition 15. *For a graph $G = (V, E)$ and vector $[r]_{i=1}^{n}$ such that $1 \leq r_i \leq deg_G(i)$, a subset of vertices D is called the generalized dominating set corresponding to r, or r-GDS, if for each vertex i, either $i \in D$ or at least r_i number of neighboring vertices are in D. The minimum generalized dominating set (MGDS) problem is to find an r-GDS of minimum size in G.*

Roman et al. [87] study the Parametrized algorithm for GDS. HARANT et al [49] introduce a function $f_r(P)$ and show that $\gamma_r(G) = \min f_r(P)$ where the minimum is taken over the $n-$dimensional cube $C^n = \{P = (p_1, \ldots, p_n) | p_i \in \Re, 0 \leq p_i \leq 1, i = 1, \ldots, n\}$.

Note that other types of greater domination are imposed on k-dominating set. A vertex $i \in V - D$ is k-dominated if it has at least k neighboring vertices in D. If all vertices in $V - D$ are k-dominated then D is called k-dominating set. The minimum k-dominating set (MkDS) problem seeks to find the smallest k-dominating set. Figure 5.1 shows instances of MDS, MSDS, MGSDS, MMSDS, MGDS, and M2DS problem in 5.1a, 5.1b, 5.1c, 5.1d, 5.1e and 5.1f respectively. In the figure, numbers on each vertex represent the domination requirement. For all problems considered in this chapter, we assume that the domination requirement is given as input parameter, which is determined considering the relative importance of sites that each vertex represents.

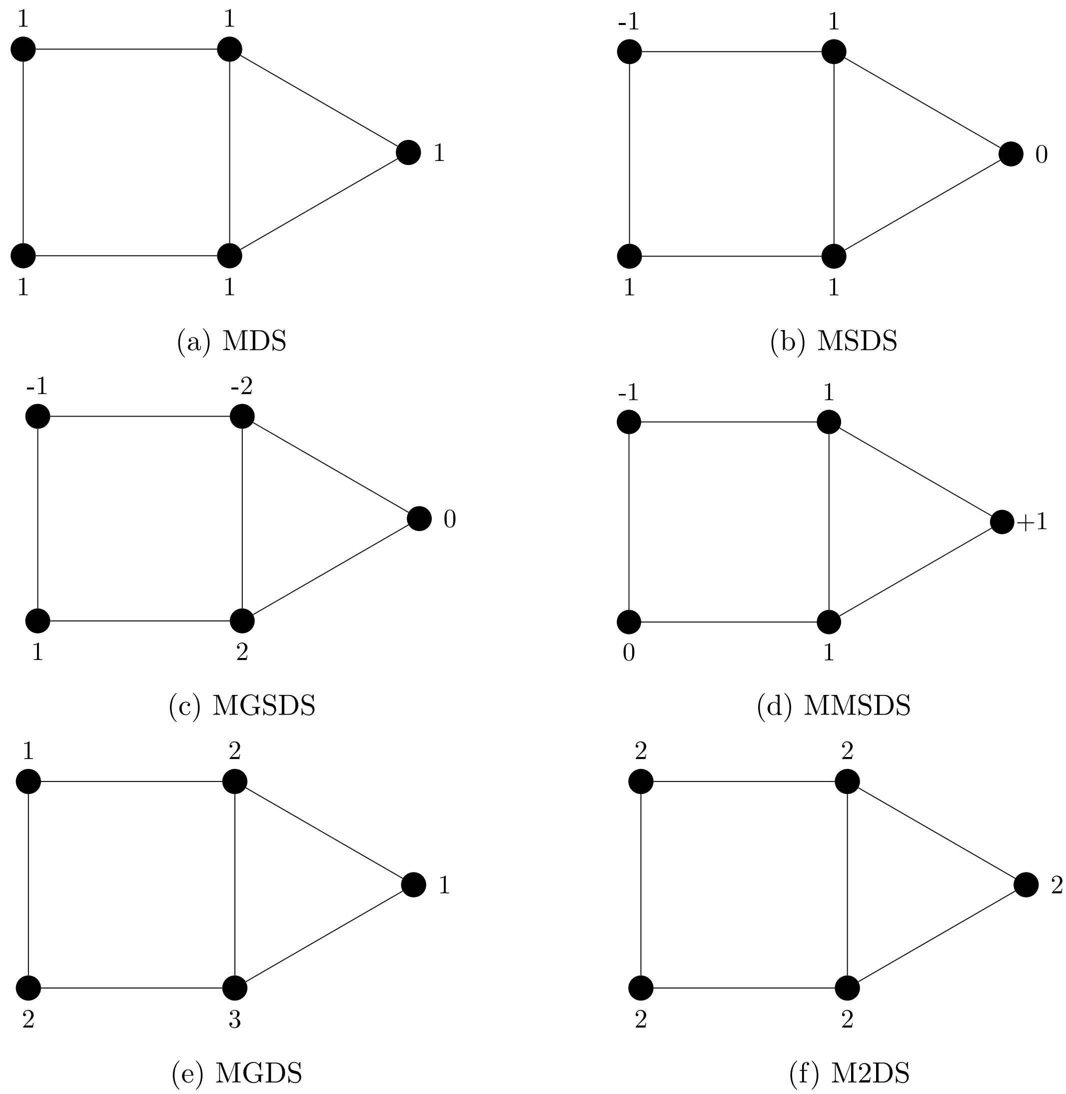
This chapter is organized in the following way. In section 5.2, we demonstrate the computational complexity of decision version of the problems introduced above. In some variations of the problems, we show that they are hard not only because they are a generalization of DOMINATING SET, but they are hard in their own respect. Section 5.3 presents the mathematical formulations and some basic polyhedral properties. Section 5.4 shows some analytical bounds on the size of the r-GDS. Section 5.5 displays some numerical experiment results of solving the problems using CPLEX 12.1® on random unit disk graphs and also on some benchmark instances. Performance of simulated annealing is also presented as against CPLEX 12.1®. Throughout this chapter, we suppose that graphs are connected. For a disconnected graph, we can tackle each connected component separately.

5.2 Computational complexity

In this section, we establish NP-completeness of the decision versions of the MSDS, MGSDS, MMSDS, and MGDS problems. Moreover, we also show the direct proof of the NP-completeness of k-DOMINATING SET for any fixed positive integer k ($2 \leq k \leq \delta(G)$). Given a graph $G = (V, E)$ with domination requirement vector r and a positive integer k, the SELECTIVE DOMINATING SET (GENERALIZED SELECTIVE DOMINATING SET) is to check whether there exists an r-SDS (r-GSDS) of size $\leq k$ in G. An instance of each of the two problems is given by a triple $\langle G = (V, E), r, k \rangle$.

Theorem 11. SELECTIVE DOMINATING SET *problem is NP-complete.*

Proof. The problem is clearly in NP. To complete the proof, we use a reduction from the classical 3-SAT problem [43]. We use a construction similar to the one proposed in [85] to establish the NP-completeness of finding degree-constrained sub-



(a) MDS (b) MSDS

(c) MGSDS (d) MMSDS

(e) MGDS (f) M2DS

Figure 5.1: Instances of various dominating set problems.

graphs. Given an instance of 3-SAT with conjunctive normal form (CNF) $F = C_1 \wedge C_2 \wedge \ldots \wedge C_m$ with variables $x_1, x_2, \ldots, x_n$ and at most three variables in each clause $C_i, i = 1, \ldots, m$, we construct an instance $\langle G_F = (V_F, E_F), r, n\rangle$ of Selective Dominating Set in polynomial time such that F is satisfiable if and only if G_F has a r-SDS of size n. To construct G_F, let

$$V_F = \cup_{i=1}^{n}\{x_i^+, x_i^0, x_i^-\} \cup_{j=1}^{m} \{c_m\};$$

$$E_F = \cup_{i=1}^{n}\{(x_i^+, x_i^0), (x_i^0, x_i^-)\} \cup_{i,j}\{(x_i^+, c_j) : C_j \text{ contains } x_i\} \cup_{i,j}\{(x_i^-, c_j) : C_j \text{ contains } \bar{x}_i\}.$$

For each vertex in V_F, we assign the domination requirement values as follows:

$$r_{x_i^+} = r_{x_i^-} = 0, r_{x_i^0} = 1 \text{ or } -1, i = 1, \ldots, n;\ r_{c_j} = 1, j = 1, \ldots, m.$$

We obtain an instance $\langle G_F = (V_F, E_F), r, n\rangle$ of the Selective Dominating Set problem. Figure 5.2 shows an illustration of this construction. Note that the construction can be completed in polynomial time. Now we have to show that F is satisfiable if and only if G_F has a r-SDS of size n.

First, suppose that F has satisfiable Truth assignment. We make a set D of vertices in G_F in the following manner. If $x_i = True$, then put the vertex x_i^+ in D, and if $x_i = False$, then put the vertex x_i^- in D. The set D is a r-SDS of G_F because of the following two reasons: (i) each x_i^0 is dominated by either x_i^+ or x_i^-, and (ii) each clause vertex C_j is dominated by at least one vertex in D since by assumption each C_j contains at least one variable with Truth assignment and, by the above construction, the corresponding vertex is in D. Therefore, there is an r-SDS of size n in G_F.

We now establish the other direction stating that if G_F has a r-SDS of size n,

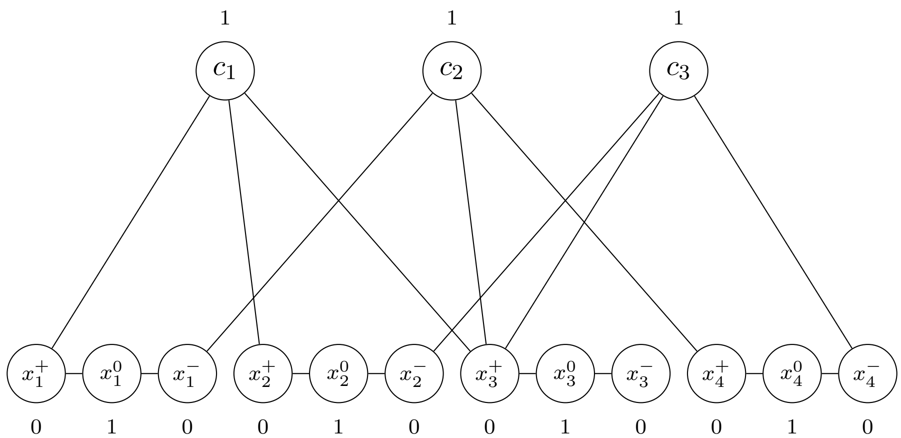


Figure 5.2: Illustration of the construction of a graph G_F when $F = C_1 \wedge C_2 \wedge C_3$, where $C_1 = (x_1 \vee x_2 \vee x_3), C_2 = (\bar{x}_1 \vee x_3 \vee x_4)$, and $C_3 = (\bar{x}_2 \vee x_3 \vee \bar{x}_4)$. G_F has SDS of size 4 if and only if F is satisfiable.

then F is satisfiable. Let D be a r-SDS in G_F of size $n = |D|$. Note that because each vertex x_i^0 must be either in D or be dominated by a vertex in D, at least one vertex from x_i^+, x_i^0, x_i^- must be in D for each $i = 1, \ldots, n$. In fact, exactly one vertex from x_i^+, x_i^0, x_i^- must be in D. Therefore, D contains no clause vertex. Because D is a r-SDS, however, each clause vertex must be dominated by at least one vertex in D. The following truth assignment makes F satisfiable : for each variable x_i, assign x_i True if $x_i \in D$, otherwise assign x_i False. It is easy to see that this makes F satisfiable. $\square$

Theorem 12. Generalized Selective Dominating Set *problem is NP-complete.*

Proof. Obviously, Generalized Selective Dominating Set is NP-complete directly from Theorem 11, since GSDS generalizes SDS. However, we aim to show the NP-completeness of the problem for an arbitrarily reasonable choice of the domination requirement vector. Namely, the proof below demonstrates that, no matter how we select the domination requirement vector $\hat{r}$ for a given graph G, G and

$\hat{r}$ can be extended to obtain a larger instance of the GENERALIZED SELECTIVE DOMINATING SET problem, which, if solved, yields a solution to the SELECTIVE DOMINATING SET problem in G. The problem is clearly in NP. Given an instance $\langle G = (V, E), r, k \rangle$ of SELECTIVE DOMINATING SET and an arbitrary domination requirement vector $\hat{r} \in \mathcal{R}_G$ such that the sign of r_i and $\hat{r}_i$ is same $\forall i$, we construct an instance $\langle G' = (V', E'), r', k' \rangle$ of GENERALIZED SELECTIVE DOMINATING SET in polynomial time such that G has a r-SDS of size k if and only if G' has a r'-GSDS of size k'. To construct G', let

$$V^{(i)} = \{v_j^{(i)} : j = 1, \ldots, |\hat{r}_i| - 1\},$$

and let $\bar{G} = (\bar{V}, \bar{E})$ be an arbitrary graph with the set of vertices $\bar{V} \equiv \cup_{i=1}^{n} V^{(i)}$. We construct a graph $G' = (V', E')$, where

$$V' = V \cup \bar{V} \cup \{a\}$$

and

$$E' = E \cup \bar{E} \cup (\cup_{i=1}^{n} \{(i, v), (a, v) : v \in V^{(i)}\}).$$

For an arbitrary non-negative vector $\bar{r} \in \mathcal{R}_{\bar{G}}$, we also define a domination requirement vector r' for G' as follows:

$$r'_i = \begin{cases} \hat{r}_i, & \text{if } i \in V; \\ \bar{r}_i, & \text{if } i \in \bar{V}; \\ -|\bar{V}|, & \text{if } i = a. \end{cases}$$

Setting $k' = k + \sum\limits_{i=1}^{n} (|r_i| - 1)$ completes the reduction and this can be done in

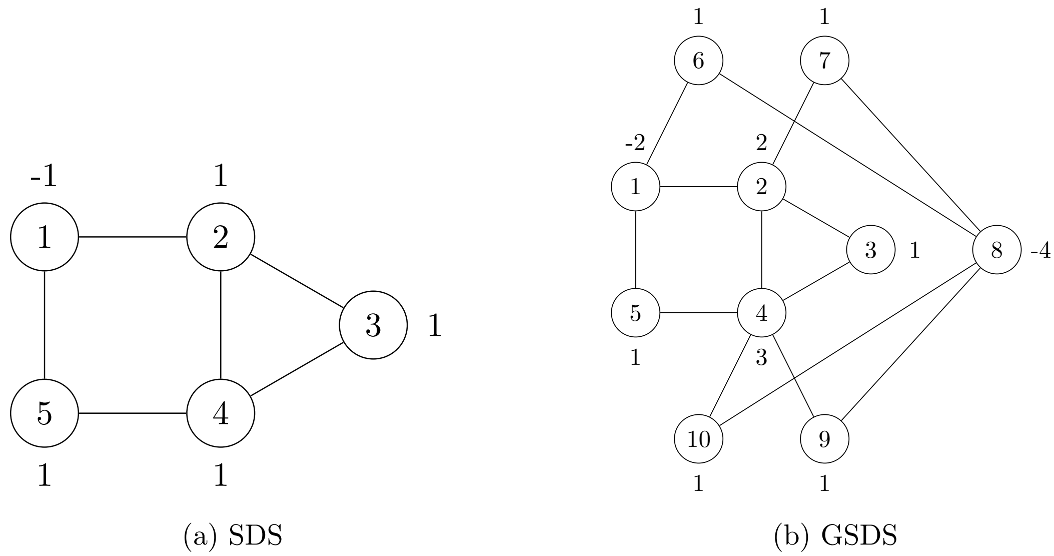


Figure 5.3: Illustration of reduction for Generalized Selective Dominating Set.

polynomial time. Figure 5.3 shows the construction for $r = \{-1, 1, 1, 1, 1\}$ and $\hat{r} = \{-2, 2, 1, 3, 1\}$.

First, we show that if there exists a r-SDS of size k in G, then G' has a r'-GSDS of size k'. Let $D \subset V$ be a r-SDS of size $k = |D|$ in G. Since $r'_a = -|\bar{V}|$, $D' = D \cup \bar{V}$ dominates a. In fact, D' dominates all vertices in V' and we have a r'-GSDS of size $k' = |D'| = k + \sum_{i=1}^{n}(|r_i| - 1)$.

We now establish the other direction by saying that if G' has a r'-GSDS of size k', then G has a r-SDS of size k. Let D' be a r'-GSDS of size k'. Since $r'_a = -|\bar{V}|$, $a \notin D'$ and $\bar{V} \in D'$. By construction, it is straightforward to see that $D = D' - \bar{V}$ is a r-SDS of size $k = k' - \sum_{i=1}^{n}(|r_i| - 1)$. $\square$

This complexity result demonstrates that Generalized Selective Dominating Set is hard not only because it is a generalization of Selective Dominating Set, but it is hard in its own respect. The next theorem demonstrates the NP-completeness of MIXED SELECTIVE DOMINATING SET problem.

Theorem 13. MIXED SELECTIVE DOMINATING SET *problem is NP-complete.*

Proof. Obviously, this problem belongs to NP since we can verify a "yes" instance in polynomial time. The reduction is similar to Theorem 11. Given an instance of 3-SAT with conjunctive normal form (CNF) $F = C_1 \wedge C_2 \wedge \ldots \wedge C_m$ with variables $x_1, x_2, \ldots, x_n$ and at most three variables in each clause $C_i, i = 1, \ldots, m$, we construct an instance $\langle G_F = (V_F, E_F), r, k\rangle$ of MIXED SELECTIVE DOMINATING SET in polynomial time such that F is satisfiable if and only if G_F has a r-MSDS of size k. The construction is exactly the same as Theorem 11 except in regards to domination requirement assignment. For each vertex in V_F, we assign the domination requirement values as follows:

$$r_{x_i^+} = r_{x_i^-} = 0, r_{x_i^0} = 1 \text{ or } -1, i = 1, \ldots, n;\ r_{c_j} = +1, j = 1, \ldots, m.$$

By letting $k = n+m$, we obtain an instance $\langle G_F = (V_F, E_F), r, k\rangle$ of the MIXED SELECTIVE DOMINATING SET problem. Note that the construction can be completed in polynomial time. Now we have to show that F has a truth assignment if and only if G_F has a r-MSDS of size k.

First, suppose that F has satisfiable truth assignment. We make a set D of vertices in G_F as follows : put all clause vertices C_j, $j = 1, \ldots, m$ in D and if $x_i = True$, then put x_i^+ in D, and if $x_i = False$, then put the vertex x_i^- in D. The set D is a r-MSDS of G_F because of the following two reasons: (i) each x_i^0 is dominated by either x_i^+ or x_i^-, and (ii) the domination requirement of each clause vertex C_j is satisfied because each C_j is in D and, by assumption, each C_j contains at least one variable with truth assignment and, by construction, the corresponding vertex is in D. Therefore, G_F has a r-MSDS of size $k = n + m$.

We now establish the other direction by stating that if G_F has a r-MSDS of size k, then F is satisfiable. Let D be a r-MSDS in G_F of size $k = n + m$. Then, since $r_{c_j} = +1$ for $j = 1 \ldots, m$, every clause vertex must be in D. Also, each clause vertex C_j $\forall j$ and x_i^0 $\forall i$ must be dominated by at least one vertex in D. This implies that for each i either x_i^+ or x_i^- must be in D. The following truth assignment makes F satisfiable : for each variable x_i, assign x_i True if $x_i \in D$, otherwise assign x_i False. It is straightforward to see that this makes F satisfiable. □

Now we show that GENERALIZED DOMINATING SET problem is NP-Complete for arbitrary graphs. Let the vertex where a leaf vertex is connected be a *stem vertex*.

Lemma 4. *Consider a graph $G = (V, E)$ with domination requirement r such that each stem vertex has only one leaf vertex. If G has a GDS, then G has a GDS including all stem vertices and excluding all leaf vertices.*

Proof. Let $D \subset V$ be a GDS in G. Then, for each stem $v \in V$ and leaf $u \in V$ pair, we have the following two cases.

1. $v \in D, u \notin D$.

2. $v \notin D, u \in D$.

If all stem and leaf vertex pair in G satisfy Case 1, then we are done. If there is a stem and leaf pair satisfying Case 2, we remove u from D and put v into D. □

From Lemma 4, we know that if G has an optimal GDS, then it has an optimal GDS including all stem vertices and excluding all leaf vertices.

Theorem 14. GENERALIZED DOMINATING SET *problem is NP-Complete.*

Proof. Obviously, the statement follows directly from the fact that GENERALIZED DOMINATING SET generalizes DOMINATING SET, which is a well known NP-complete problem. However, we present more direct proof for an arbitrarily reasonable choice of the domination requirement vector with $1 \le r_i \le d(i)\ \forall i \in V$. Namely, this proof demonstrates that no matter how we select the domination requirement vector $\hat{r}$ for a given graph G, G and $\hat{r}$ can be extended to obtain a larger instance of the GENERALIZED DOMINATING SET problem, which, if solved, yields a solution to the DOMINATING SET problem in G. This clearly belongs to NP. Given an instance $\langle G = (V, E), r, k\rangle$, $r_i = 1\ \forall i \in V$, of DOMINATING SET, we construct an instance $\langle G' = (V', E'), r', k'\rangle$ in polynomial time such that G has a dominating set of size k if and only if G' has a generalized dominating set of size k'. To construct G', let

$$V_1^{(i)} = \{v_j^{(i)} : j = 1, \ldots, |\hat{r}_i| - 1\}\ , V_2^{(i)} = \{u_j^{(i)} : j = 1, \ldots, |\hat{r}_i| - 1\}$$

and

$$\bar{E} = \{(v_j^{(i)}, u_j^{(i)}) : i = 1, \ldots, n, j = 1, \ldots, |\hat{r}_i| - 1\}.$$

We construct a graph $G' = (V', E')$, where

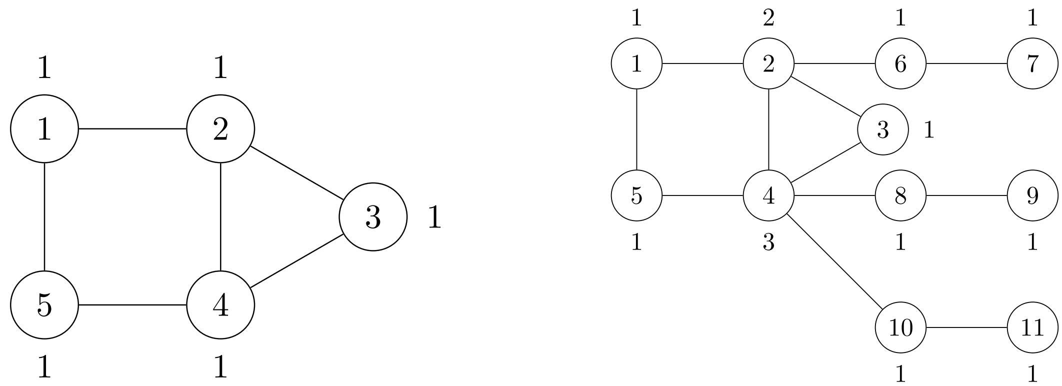
$$V' = V \cup (\cup_{i=1}^{n} V_1^{(i)}) \cup (\cup_{i=1}^{n} V_2^{(i)})$$

and

$$E' = E \cup \bar{E} \cup (\cup_{i=1}^{n}\{(i, v) : v \in V_1^{(i)}\}).$$

We also define a domination requirement vector r' for G' as follows:

$$r'_i = \begin{cases} \hat{r}_i, & \text{if } i \in V; \\ 1, & \text{o.w.} \end{cases}$$



(a) DS (b) GDS

Figure 5.4: Illustration of reduction for Generalized Dominating Set.

Setting $k' = k + |\cup_{i=1}^{n} V_1^{(i)}|$ completes the reduction. Note that this reduction can be done in polynomial time. Figure 5.4 shows the construction example for $r = \{1,1,1,1,1\}$ and $\hat{r} = \{1,2,1,3,1\}$.

We now show that if there exists a dominating set of size k in G, then G' has a generalized dominating set of size k'. Let $D \subset V$ be a dominating set of size $k = |D|$ in G. Then it is clear that $D' = D \cup (\cup_{i=1}^{n} V_1^{(i)})$ dominates all vertices in G'. For the other direction, let D' be a generalized dominating set of size k' in G' such that D' includes all stem vertices and does not include any leaf vertex. The existence of such D' directly follows from Lemma 4. Then it is clear that $D = D' \setminus (\cup_{i=1}^{n} V_1^{(i)})$ is a dominating set of size $k = |D|$ for G. □

The complexity result demonstrates that Generalized Dominating Set is hard not only because it is a generalization of Dominating Set, but also it is a hard problem in its own respect.

For a graph G with $\delta(G) \geq k \geq 2$, if $r_i = k\ \forall i \in V$, then we have a special case of MGDS problem called Minimum k-dominating set problem. To the best of my knowledge, there is no known hardness proof for an arbitrary value of $k \geq 2$ for this problem. Note that the reduction shown in Theorem 14 does not apply, since there are leaf vertices with domination requirement 1 in that reduction.

Theorem 15. k-DOMINATING SET *problem is NP-complete for any fixed positive integer* k $(2 \leq k \leq \delta(G))$.

Proof. Obviously, the statement follows directly from the fact that k-Dominating Set generalizes Dominating Set. However, we provide more direct proof for any positive integer k $(2 \leq k \leq \delta(G))$. Namely, we demonstrate that no matter how we select the integer $k \geq 2$ for a given graph G, G can be extended to obtain a larger instance of the k-Dominating Set problem, which, if solved, yields a solution to the Dominating Set problem. This problem is clearly in NP. Given an instance $\langle G = (V, E), c\rangle$ of Dominating Set, we construct an instance $\langle G' = (V', E'), c'\rangle$ of k-Dominating Set in polynomial time such that G has a dominating set of size c if and only if G' has a k-dominating set of size c'. To construct G', we expand G by adding $(k-1)$ copies of complete graph of order $k+3$ for each vertex in G (i.e. we add n(k-1) complete graphs of order k+3 in total). Denote the r^{th} $(r \in \{1, \ldots, k-1\})$ such copy added for $i \in V$ by V_{i^r}, where

$$V_{i^r} = \{1_{i^r}, 2_{i^r}, \ldots, (k+3)_{i^r}\}$$

. We construct a graph $G' = (V', E')$, where $V' = V \cup (\cup_{i \in V, r \in \{1,\ldots,k-1\}} V_{i^r})$ and

$$E' = E \cup (\cup_{i \in V, r \in \{1,\ldots,k-1\}} E_{i^r}) \cup \tilde{E}$$

where

$$E_{i^r} = \{(j, l) | j, l \in V_{i^r}\}$$

and

$$\tilde{E} = \{(1_{i^r}, i) | i \in V, r \in \{1, \ldots, k-1\}\}.$$

The set E_{i^r} represents the edges between each pair of vertices in V_{i^r}. The set $\tilde{E}$ includes the cross edges between each vertex $i \in V$ and one vertex in each $k-1$ clique copies. For instance, vertex $1 \in V$ is adjacent to vertices $1_{1^1}, \ldots, 1_{1^{k-1}}$ and vertex $2 \in V$ is adjacent to $1_{2^1}, \ldots, 1_{2^{k-1}}$ and so on. If G is connected, by this construction we know $\delta(G') \geq k$ since vertex $j \in V_{i^r}$ has $deg_{G'}(j) \geq k+2$ and $i \in V$ has $deg_{G'}(i) \geq k$. Putting $c' = c + nk(k-1)$ and assigning $r_i' = k \ \forall i \in V'$ completes the reduction. Note that the instance $\langle G' = (V', E'), c' \rangle$ can be constructed in polynomial time.

It is easy to see that if there exists a dominating set of size c in G, then G' has a k-dominating set of size c' and vice versa. □

This complexity result shows that k-Dominating Set is hard not only because it is a generalization of Dominating Set, but also it is a hard problem in its own respect.

5.3 IP formulations and polytopes

In this section, we demonstrate basic polyhedral properties of dominating set polytope, r-SDS polytope, r-GSDS polytope, r-MSDS polytope, and r-GDS polytope. For notational convenience, we use the following.

- $V^{<} \subset V$: the set of vertices with $r_i < 0$
- $V^{=} \subset V$: the set of vertices with $r_i = 0$
- $V^{>} \subset V$: set of vertices with $r_i > 0$

In each formulation, the n-vector x of 0-1 variables is the characteristic vector of each variation of dominating set for a graph $G = (V, E)$. Let $\mathbf{e}_i$ be the unit vector with i^{th} component 1 and the rest 0; $\sum_{i=1}^{n} \mathbf{e}_i = \mathbf{e}_1 + \ldots + \mathbf{e}_n$. We denote the $n \times n$ identity matrix by I.

5.3.1 Dominating set

Given a graph $G = (V, E)$ with $V = \{1, \ldots, n\}$ and the adjacency matrix A of G, the following binary integer programming formulation can be used to solve minimum dominating set problem:

$$
\begin{aligned}
\text{minimize} \sum_{i=1}^{n} x_i & & \\
\text{s.t. } (A+I)x & \geq & 1, \\
x_i & \in & \{0,1\},\ i \in \{1, 2, \ldots, n\}.
\end{aligned}
$$

The dominating set polytope $P(G)$ is given by

$$P(G) = conv(\{x \in \{0,1\}^n | (A+I)x \geq 1\}).$$

The following theorem establishes the basic properties.

Theorem 16. *Let $P(G)$ denote the dominating set polytope of a graph $G = (V, E)$. Then, (1) $dim(P(G)) = n$; (2) $x_i \geq 0$ induces a facet if $deg_G(u) > 1\ \forall u \in N[i]$; (3) $x_i \leq 1$ induces facets of $P(G)$ for every $i \in V$*

Proof. Since the set of vertices V itself is a dominating set, the point $\sum_{i=1}^{n} e_i$ is in $P(G)$. Any vertex $u \in V$ can be dominated by its neighboring vertex. Thus, the n points $\sum_{i=1, i \neq j}^{n} e_i$ for all $j \in V$ are in $P(G)$. Clearly these $n+1$ points are affinely independent and we obtain $dim(P(G)) = n$.

Let $F = \{x \in P(G)|x_i = 0\}$ for which $deg_G(u) > 1 \ \forall u \in N[i]$. Obviously $\sum_{j=1,j\neq i}^{n} e_j$ is in F since $deg_G(i) > r_i$. When two vertices $i, k \in N[i]$ are not in the dominating set, the domination requirements r_i and r_k can be satisfied by $V \setminus \{i, k\}$ since $deg_G(i) > 1$ and $deg_G(k) > 1$. Thus we have the points $\sum_{j=1,j\neq i,j\neq k}^{n} e_j$ for all $k \in V \setminus \{i\}$ in F. These n points are clearly affinely independent and $dim(F) = n-1$. Thus, it is a facet.

Let $F = \{x \in P(G)|x_i = 1\}$. We first observe that any vertex set $V \setminus \{j\}$ such that $j \neq i$ is a dominating set (i.e. j can be dominated by at least one of its neighboring vertices in $V \setminus \{j\}$). Thus we have $\sum_{j=1}^{n} e_j$ and $\sum_{j=1,j\neq k}^{n} e_j$ for all $k \in V \setminus \{i\}$ in F and these are n affinely independent points. Thus $dim(F) = n - 1$ and it is a facet. □

5.3.2 Selective dominating set

The following binary integer programming formulation can be used to solve MSDS problem:

$$
\begin{aligned}
\text{minimize} \sum_{i=1}^{n} x_i & & \quad (5.1)\\
\text{s.t. } (A + rI)x &\geq |r'|,\\
x_i &= 0, \forall i \in V^{<}\\
x_i &\in \{0, 1\},\ i \in \{1, 2, \ldots, n\}.
\end{aligned}
$$

where $|r'|$ is a vector of absolute values of each element in r where $r'_i = 0 \ \forall i \in V^{=}$. The SDS polytope $P(G)$ is given by

$$P(G) = conv(\{x \in \{0, 1\}^n |(A + rI)x \geq |r'|, x_i = 0 \ \forall i \in V^{<}\}).$$

Theorem 17. *Let $P(G)$ denote the selective dominating set polytope of $G = (V, E)$ with given $r \in \{-1, 0, 1\}^n$. If $d_r^+(i) > 1$ for all $i \in V \setminus V^=$ then, (1) $dim(P(G)) = n - |V^<|$; (2) $x_u \leq 1$ induces facets of $P(G)$ for every $u \in V \setminus V^<$.*

Proof. We know that $x_i = 0$ if $r_i = -1$ since the vertex $i \in V$ cannot be included in r-SDS and we have $|V^<|$ linearly independent equalities and $dim(P(G)) \leq n - |V^<|$. We claim that there exist $n - |V^<| + 1$ affinely independent points in $P(G)$. Since we assumed that G has a feasible r-SDS, $\sum_{i \in V \setminus V^<} e_i$ is in $P(G)$ (i.e. vectors with $x_i = 1\ \forall i \in V \setminus V^<$ and $x_i = 0\ \forall i \in V^<$ are in $P(G)$). If $d_r^+(i) > 1$ for all $i \in V \setminus V^=$ then any set of vertices $V \setminus (V^< \cup \{k\})$ for each $k \in V \setminus V^<$ is a feasible r-SDS. Thus we have $\sum_{i \in V \setminus V^<, i \neq k} e_i$ for each $k \in V \setminus V^<$ in $P(G)$. It is clear that these points $\sum_{i \in V \setminus V^<} e_i$ and $\sum_{i \in V \setminus V^<, i \neq k} e_i$ for each $k \in V \setminus V^<$ are affinely independent. Thus we have $n - |V^<| + 1$ affinely independent points in $P(G)$ and $dim(P(G)) = n - |V^<|$.

Let $F = \{x \in P(G) | x_u = 1\}$, $u \in V \setminus V^<$. We claim that there are $n - |V^<|$ affinely independent points in F. Clearly, $\sum_{j \in V \setminus V^<} e_j$ is in F. Since $d_r^+(i) > 1$ for all $i \in V \setminus V^=$, the points $\sum_{j \in V \setminus V^<, j \neq k} e_j + e_u$ for all $k \in V \setminus (V^< \cup \{u\})$ are in F. These $n - |V^<|$ points are obviously affinely independent. Therefore, we have $dim(F) = n - |V^<| - 1$, and it is a facet. □

5.3.3 Generalized selective dominating set

The mathematical programming formulation of GSDS can be stated in the following way:

$$
\begin{aligned}
\text{minimize } & \sum_{i=1}^{n} x_i \qquad\qquad (5.2)\\
\text{s.t. } (A+rI)x &\geq |r'|,\\
x_i &= 0, \forall i \in V^{<}\\
x_i &\in \{0,1\},\ i \in \{1,2,\ldots,n\}.
\end{aligned}
$$

where $|r'|$ is a vector of absolute value of each element in r. The GSDS polytope $P(G)$ is given by

$$P(G) = conv(\{x \in \{0,1\}^n | (A+rI)x \geq |r'|, x_i = 0\ \forall i \in V^{<}\}).$$

Theorem 18. *Let $P(G)$ denote the generalized selective dominating set polytope of $G = (V,E)$ with given $r \in \mathcal{R}_G$. If $d_r^+(i) > |r_i|$ for all $i \in V \setminus V^{=}$, then (1) $dim(P(G)) = n - |V^{<}|$; (2) $x_u \leq 1$ induces facets of $P(G)$ for every $u \in V \setminus V^{<}$.*

Proof. We know that $x_i = 0$ if $r_i < 0$ since the vertex $i \in V$ cannot be included in r-GSDS and we have $|V^{<}|$ linearly independent equalities. Thus, $dim(P(G)) \leq n - |V^{<}|$. We claim that there exist $n - |V^{<}| + 1$ affinely independent points in $P(G)$. Since we assumed that G has a feasible r-GSDS, $\sum_{i \in V \setminus V^{<}} e_i$ is in $P(G)$ (i.e. vectors with $x_i = 1\ \forall i \in V \setminus V^{<}$ and $x_i = 0\ \forall i \in V^{<}$ are in $P(G)$). If $d_r^+(i) > |r_i|$ for all $i \in V \setminus V^{=}$ then any set of vertices $V \setminus (V^{<} \cup \{k\})$ for each $k \in V \setminus V^{<}$ is a feasible r-GSDS. Thus we have $\sum_{i \in V \setminus V^{<}, i \neq k} e_i$ for each $k \in V \setminus V^{<}$ in $P(G)$. It is clear that these points $\sum_{i \in V \setminus V^{<}} e_i$ and $\sum_{i \in V \setminus V^{<}, i \neq k} e_i$ for each $k \in V \setminus V^{<}$ are affinely independent. Thus we have $n - |V^{<}| + 1$ affinely independent points in $P(G)$ and $dim(P(G)) = n - |V^{<}|$.

Let $F = \{x \in P(G) | x_u = 1\}$, $u \in V \setminus V^{<}$. We claim that there are $n - |V^{<}|$

affinely independent points in F. Clearly, $\sum_{j\in V\setminus V^<} e_j$ is in F because we assumed that G has a feasible solution. Since $d_r^+(i) > |r_i|$ for all $i \in V \setminus V^=$, the points $\sum_{j\in V\setminus V^<, j\neq k} e_j$ for all $k \in V\setminus V^< \setminus \{u\}$ are in F. These $n-|V^<|$ points are obviously affinely independent. Thus we have $dim(F) = n - |V^<| - 1$, and it is a facet. □

5.3.4 *Mixed selective dominating set*

Let $V^{+1} = \{i \in V | r_i = +1\}$. Then the following binary integer programming formulation can be used to solve MMSDS problem:

$$
\begin{aligned}
\text{minimize} \sum_{i=1}^{n} x_n & & \\
\text{s.t. } (A + rI)x &\geq |r'|, \\
x_i &= 0, \forall i \in V^< \\
x_i &= 1, \forall i \in V^{+1} \\
x_i &\in \{0,1\},\ i \in \{1,2,\dots,n\}.
\end{aligned}
$$

where $|r'|$ is a vector of absolute value of each element in r and $r'_i = 2\ \forall i \in V^{+1}$. Note that $r'_i = 2\ \forall i \in V^{+1}$ is to satisfy the domination requirement of $i \in V^{+1}$. The MSDS polytope $P(G)$ is given by

$$P(G) = conv(\{x \in \{0,1\}^n | (A + rI)x \geq |r'|, x_i = 0\ \forall i \in V^<, x_i = 1\ \forall i \in V^{+1}).$$

Theorem 19. *Let $P(G)$ denote the mixed selective dominating set polytope of $G = (V,E)$ with given $r \in \{-1,0,1,+1\}^n$ and $\bar{d}_r^+(i) = |\{j \in N(i)|r_j \in \{0,1\}\}|$ for $i \in V$. If $\bar{d}_r^+(i) > 1$ for all $i \in V \setminus V^=$ then, (1) $dim(P(G)) = n - |V^< \cup V^{+1}|$; (2) $x_u \leq 1$ induces facets of $P(G)$ for every $u \in V \setminus (V^< \cup V^{+1})$.*

Proof. We know that $x_i = 0$ if $r_i = -1$ since the vertex $i \in V$ cannot be included in

r-MSDS and $x_i = 1$ if $r_i = +1$ since the vertex $i \in V$ must be in r-MSDS. We have $|V^< \cup V^{+1}|$ linearly independent equalities and $dim(P(G)) \leq n - |V^< \cup V^{+1}|$. We claim that there exist $n - |V^< \cup V^{+1}| + 1$ affinely independent points in $P(G)$. Since we assumed that G has a feasible r-MSDS, $\sum_{i \in V - V^<} e_i$ is in $P(G)$ (i.e. vectors with $x_i = 1\ \forall i \in V \setminus V^<$ and $x_i = 0\ \forall i \in V^<$ are in $P(G)$). If $\bar{d}_r^+(i) > 1$ for all $i \in V \setminus V^=$ then any set of vertices $V \setminus (V^< \cup \{k\})$ for each $k \in V \setminus (V^< \cup V^{+1})$ is feasible r-MSDS. Thus we have $\sum_{i \in V \setminus V^<, i \neq k} e_i$ for each $k \in V \setminus (V^< \cup V^{+1})$ in $P(G)$. It is clear that these points $\sum_{i \in V \setminus V^<} e_i$ and $\sum_{i \in V \setminus V^<, i \neq k} e_i$ for each $k \in V \setminus (V^< \cup V^{+1})$ are affinely independent. Thus we have $n - |V^< \cup V^{+1}| + 1$ affinely independent points in $P(G)$ and $dim(P(G)) = n - |V^< \cup V^{+1}|$.

Let $F = \{x \in P(G) | x_u = 1\}$, $u \in V \setminus (V^< \cup V^{+1})$. We claim that there are $n - |V^< \cup V^{+1}|$ affinely independent points in F. Clearly, $\sum_{j \in V \setminus V^<} e_j$ is in F because we assumed that G has feasible r-MSDS. Since $\bar{d}_r^+(i) > 1$ for all $i \in V \setminus V^=$, the points $\sum_{j \in V \setminus V^<, j \neq k} e_j$ for all $k \in V \setminus (V^< \cup V^{+1} \cup \{u\})$ are in F. These $n - |V^< \cup V^{+1}|$ points are obviously affinely independent. Thus we have $dim(F) = n - |V^< \cup V^{+1}| - 1$ and it is a facet. $\square$

5.3.5 *Generalized dominating set ($[r_i]_{i=1}^n$ $1 \leq r_i \leq d(i)$)*

The following binary integer programming formulation can be used to solve the MGDS problem:

$$\begin{aligned} \text{minimize} \sum_{i=1}^{n} x_i & \\ \text{s.t. } (A + rI)x &\geq r, \\ x_i &\in \{0,1\},\ i \in \{1, 2, \ldots, n\}. \end{aligned}$$

where $1 \leq r_i \leq d(i)\ \forall i \in V$. The GDS polytope $P(G)$ is given by

$$P(G) = conv(\{x \in \{0,1\}^n | (A + rI)x \geq r\})$$

Theorem 20. *Let $P(G)$ denote the GDS polytope of a graph $G = (V, E)$ with domination requirement $[r_i]_{i=1}^n$ such that $1 \leq r_i \leq d(i)\ \forall i \in V$. Then, (1) $dim(P(G)) = n$; (2) $x_i \geq 0$ induces a facet of $P(G)$ if $deg_G(u) > r_u\ \forall u \in N[i]$; (3) $x_i \leq 1$ induces a facet of $P(G)$ for every $i \in V$.*

Proof. Obviously, the point $\sum_{i=1}^n e_i$ is in $P(G)$ since the vertex set V itself is a r-GDS. Since $1 \leq r_i \leq deg_G(i)\ \forall i \in V$, $V \setminus \{i\}$ dominates vertex i (i.e. the vertex i has $deg_G(i)$ neighbors in $V \setminus \{i\}$ and the domination requirement of i is satisfied). Thus for any $i \in V$, the vertex set $V \setminus \{i\}$ is a r-GDS. Then we have n points $\sum_{i=1, i\neq j}^n e_i$ for each $j \in V$ in $P(G)$. Clearly these $n+1$ points are affinely independent. Thus $dim(P(G)) = n$.

Let $F = \{x \in P(G) | x_i = 0\}$ for which $deg_G(u) > r_u\ \forall u \in N[i]$. Obviously $\sum_{j=1, j\neq i}^n e_j$ is in F since $deg_G(i) > r_i$. For vertex i and any vertex $k \in V \setminus \{i\}$, the set of vertices $V \setminus \{i, k\}$ satisfies the domination requirement of all vertices in G since by assumption $deg_G(u) > r_u\ \forall u \in N[i]$. Thus we have the points $\sum_{j=1, j\neq i, j\neq k}^n e_j$ for every $k \in V \setminus \{i\}$ in $P(G)$. These n points are clearly affinely independent and $dim(F) = n - 1$. Thus it is a facet.

Let $F' = \{x \in P(G) | x_i = 1\}$. The vector $\sum_{j=1}^n e_j$ is obviously in F' since V itself is a r-GDS. Since $deg_G(j) > r_j\ \forall j \in V$, any vertex set $V \setminus \{j\}$ such that $j \neq i$ is a r-GDS (i.e. vertex j can be dominated by at least r_j number of its neighbors in $V \setminus \{j\}$). Thus we have $\sum_{j=1, j\neq k}^n e_j$ for every $k \in V \setminus \{i\}$ in F'. We found n points in F' and they are affinely independent. Therefore $dim(F') = n - 1$, and it is a facet. $\square$

5.4 Analytical bounds

Let the size of the smallest k-dominating set be *k-domination number*, $\gamma_k(G)$. Similarly, let the size of the smallest r-GDS be *generalized domination number* corresponding to r, $\gamma_r(G)$. Note that every k-dominating set ($k \geq 1$) is a dominating set in the usual sense; thus for every graph we have $\gamma(G) \leq \gamma_k(G)$ for each $k \geq 1$. A minimum 1-dominating set is a minimum dominating set and $\gamma(G) = \gamma_1(G)$. More generally, if $1 \leq d \leq k$, then every k-dominating set in G is also an d-dominating set and thus $\gamma_d(G) \leq \gamma_k(G)$. Similarly, every r-GDS is a dominating set in the usual sense; thus for every graph G, we have $\gamma(G) \leq \gamma_r(G)$. In particular, if $r_i = 1, \forall i \in V$, a minimum r-dominating set is a minimum dominating set and $\gamma(G) = \gamma_r(G)$. More generally, if $k \leq \min\{r_i\}$, then every r-dominating set in G is also a k-dominating set and thus $\gamma_k(G) \leq \gamma_r(G)$. If $\max\{r_i\} \leq k$, then every k-dominating set is also r-dominating set and thus $\gamma_r(G) \leq \gamma_k(G)$.

We have the following theorem from Fink and Jacobson concerning minimum dominating set problem.

Theorem 21. [36] *If D is a minimum dominating set of a graph G, then at least one vertex in $V - D$ is dominated by no more than two vertices in D.*

From this theorem, we get the following result for r-GDS.

Corollary 2. *Given a graph $G = (V, E)$ and the domination requirement $1 \leq r_i \leq deg_G(i)$ $\forall i \in V$, if $\min_{i \in V}\{r_i\} \geq 3$, then $\gamma_r(G) > \gamma(G)$.*

Proof. From Theorem 21, if $\min\{r_i\} \geq 3$, no minimum dominating set in G can be a r-GDS, since every minimum dominating set will dominate at least one vertex at most twice. $\square$

The next theorem generalizes the lower bound for k-dominating set shown in [36]. Note that Corollary 2 is also implied by this theorem.

Theorem 22. *Given a graph $G = (V, E)$ and the domination requirement $2 \le r_i \le deg_G(i)\ \forall i \in V$, if $\min_{i \in V}\{r_i\} = c$ then $\gamma_r(G) \ge \gamma(G) + c - 2$.*

Proof. Let D_r be a minimum r-GDS in G. Let $u \in V - D_r$ and $v_1, \ldots, v_{r_u} \in D_r$ be distinct vertices that dominate u. Note that the condition $d(i) \ge r_i\ \forall i \in V$ implies that $V - D_r \ne \emptyset$ since there is always an r-GDS which does not contain a vertex i such that $d(i) = \Delta(G)$. Since D_r is an r-GDS with $r_i \ge c \ge 2\ \forall i \in V$, each vertex $i \in V - D_r$ is dominated by at least c vertex in D_r. In other words, each vertex in $V - D_r$ is dominated by at least one vertex in $D_r - \{v_2, \ldots, v_c\}$. Therefore, since u dominates each vertex in $v_2, \ldots, v_c$, we know that $D = D_r - \{v_2, \ldots, v_c\} \cup \{u\}$ is a dominating set in G. Thus $\gamma(G) \le |D| = \gamma_r(G) - (c - 1) + 1 = \gamma_r(G) - c + 2$. □

Although Theorem 22 gives a lower bound on $\gamma_r(G)$, it is still difficult to get the lower bound because $\gamma(G)$ is difficult to obtain since the minimum dominating set problem is NP-complete also. In [36], Fink and Jacobson show a lower bound of $\gamma_k(G)$ using the maximum degree $\Delta(G)$. By similar argument, we can show the following.

Theorem 23. *For any graph $G = (V, E)$, if $\min_{i \in V}\{r_i\} = c$, then $\gamma_r(G) \ge \frac{cn}{\Delta(G)+c}$.*

Proof. Let D_r be a minimum r-GDS and s be the number of edges between D_r and $V - D_r$. Since the degree of each vertex in D_r is at most $\Delta(G)$, $s \le \Delta(G)\gamma_r(G)$. But since each vertex in $V - D_r$ is adjacent to at least c vertices in D_r, we have $s \ge c(n - \gamma_r(G))$. Combining the two inequalities, we know that $\gamma_r(G) \ge \frac{cn}{\Delta(G)+c}$. □

The following theorem, which extends the result of γ_k in [36], gives a lower bound that can be computed using the number of vertices and edges of a graph.

Theorem 24. *For a graph $G = (V, E)$ with $|V| = n$ and $|E| = m$, if* $\min_{i \in V}\{r_i\} = c$, *then* $\gamma_r(G) \geq n - \frac{m}{c}$

Proof. Let D_r be a minimum r-GDS in G. Since each vertex in $V - D_r$ is dominated by at least c or more vertices in D_r, it follows that $m \geq c|V - D_r| = c(n - \gamma_r(G))$ and thus $\gamma_r(G) \geq n - \frac{m}{c}$ □

5.5 Numerical experiments

In this section, we present some numerical test results of the considered problems on two different classes of graphs: random unit disk graphs and DIMACS Implementation Challenge instances [31]. All computational experiments were conducted on a *Dell Precision WorkStation T7500*® computer with eight 2.40 GHz Intel Xeon® processors, and 12 GB RAM. As a solver, ILOG CPLEX 12.1® was used. In section 5.5.1, we use CPLEX 12.1® to solve MDS, MSDS, MGSDS, MMSDS and MGDS problems on random unit disk graphs. In section 5.5.2, we solve MGDS problem using simulated annealing (SA) and CPLEX 12.1® on random unit disk graphs and DIMACS instances.

5.5.1 Test results on random unit disk graphs

Several random unit disk graphs are generated such that the number of nodes and the communication thresholds are specified as sets of parameters. Given a square box of a certain size, we randomly generate points having 'x' and 'y' coordinates, where each coordinate is uniformly distributed between 0 and the length of the box. The parameters used to construct these graphs are the same as used in [19] except the number of vertices in the graphs. Instead of $n = 100$ and $n = 150$, we use $n = 500$ and $n = 800$ to create more challenging instances since the preliminary experiments with CPLEX 12.1® show $n = 100$ and $n = 150$ generate easy instances for all problems.

Each problem is solved with a time limit of $3,600$ seconds. We report the optimum objective function value (if optimal is found), solving time, and optimality gap (%). This preliminary experiment is conducted to see if different values of domination requirement impose different relative difficulty of the problem. Domination requirement for each problem is assigned in the following manner;

- MDS: $r_i = 1 \; \forall i \in V$
- MSDS: randomly set $5\%, 5\%$, and 90% of vertices to have domination requirement of $-1, 0$, and 1, respectively.
- MGSDS: randomly set $3\%, 3\%$, and 94% of vertices to have domination requirement of $< 0, 0$, and > 0, respectively. If a vertex $i \in V$ is selected to have $r_i < 0$, we set $r_i = -\lfloor 0.2d(i) \rfloor$, and similarly if vertex $i \in V$ is selected to have $r_i > 0$, we set $r_i = \lceil 0.2d(i) \rceil$.
- MMSDS: randomly set $3\%, 3\%, 3\%$, and 91% of vertices to have domination requirement of $-1, 0, +1$, and 1, respectively.
- MGDS: $r_i = \lceil 0.2d(i) \rceil \forall i \in V$.

We only consider instances with a feasible solution. Existence of a feasible solution can be easily checked by computing the open-neighborhood of each vertex. Table 5.1 and 5.2 show the experiment results on $n = 500$ and 800, respectively. For these test instances, we can see that MDS and MSDS problems are relatively easier to solve than other problems for CPLEX 12.1®. For MGSDS, MMSDS and MGDS problems there is a tendency for an instance with low density to give a larger optimality gap than a denser one.

5.5.2 Test results for simulated annealing

Sanchis et al. [90] introduce several heuristic algorithms for MDS problem and report their performances on random unit disk graphs. [54] applied SA for MDS problem and performed numerical experiments on random unit disk graphs created using the same parameters as in [90]. The experimental results in [54] demonstrated that SA solves MDS very efficiently. We apply SA to solve MGDS problem on random unit disk graphs with parameters as in [90, 54]. Table 5.3, 5.4, and 5.5 show the test results. In these Tables, the column GR shows the results of a greedy algorithm. In each iteration of GR, a vertex which could satisfy the most unsatisfied domination requirement is added to generalized dominating set. GR and SA are applied 10 times in each instance and the best, average, and standard deviations of the size of generalized dominating set are reported. In comparison with CPLEX 12.1®, the better solution is highlighted in bold.

In order to help the replication of the test result for future research, we perform numerical experiments on DIMACS Implementation Challenge instances. For instances with $n \leq 4100$, both SA and CPLEX 12.1® are applied and for instance with $n \geq 4700$ only SA is applied. Table 5.6 through 5.10 show the test results on instances with $n \leq 4100$, and Table 5.11 and 5.12 present the result for $n \geq 4700$.

Table 5.1: Test results on random unit disk graphs with $n = 500$.

			MDS			MSDS			MGSDS			MMSDS			MGDS		
dimensions	radius	$\rho(G)$	*opt*	time	gap	*opt*	time	gap	*opt*	time	gap	*opt*	time	gap	*opt*	time	gap
100×100	20	0.11	11	0	0.00	11	0	0.00	≤ 100	> 3600	0.13	≤ 100	> 3600	0.13	≤ 100	> 3600	0.13
	25	0.16	8	0	0.00	8	0	0.00	≤ 100	> 3600	0.12	≤ 100	> 3600	0.12	≤ 100	> 3600	0.12
	30	0.22	6	0	0.00	6	0	0.00	≤ 94	> 3600	0.08	≤ 94	> 3600	0.08	≤ 94	> 3600	0.08
	35	0.28	4	0	0.00	4	0	0.00	≤ 98	> 3600	0.12	≤ 98	> 3600	0.12	≤ 98	> 3600	0.12
	40	0.34	4	0	0.00	4	0	0.00	≤ 97	> 3600	0.09	≤ 97	> 3600	0.09	≤ 97	> 3600	0.09
	45	0.41	4	0	0.00	4	0	0.00	≤ 96	> 3600	0.07	≤ 96	> 3600	0.07	≤ 96	> 3600	0.07
	50	0.48	3	0	0.00	3	0	0.00	≤ 94	> 3600	0.05	≤ 94	> 3600	0.05	≤ 94	> 3600	0.05
120×120	20	0.08	16	0	0.00	16	0	0.00	≤ 102	> 3600	0.14	≤ 102	> 3600	0.14	≤ 102	> 3600	0.14
	25	0.11	11	0	0.00	11	0	0.00	≤ 101	> 3600	0.13	≤ 101	> 3600	0.13	≤ 101	> 3600	0.13
	30	0.16	8	0	0.00	8	0	0.00	≤ 100	> 3600	0.12	≤ 100	> 3600	0.12	≤ 100	> 3600	0.12
	35	0.21	6	0	0.00	6	0	0.00	≤ 93	> 3600	0.07	≤ 93	> 3600	0.07	≤ 93	> 3600	0.07
	40	0.26	5	0	0.00	5	0	0.00	≤ 97	> 3600	0.11	≤ 97	> 3600	0.11	≤ 97	> 3600	0.11
	45	0.31	4	0	0.00	4	0	0.00	≤ 97	> 3600	0.09	≤ 97	> 3600	0.09	≤ 97	> 3600	0.09
	50	0.37	4	0	0.00	4	0	0.00	≤ 96	> 3600	0.07	≤ 96	> 3600	0.07	≤ 96	> 3600	0.07
140×140	30	0.12	11	0	0.00	10	0	0.00	≤ 99	> 3600	0.12	≤ 99	> 3600	0.12	≤ 99	> 3600	0.12
	35	0.16	8	0	0.00	8	0	0.00	≤ 100	> 3600	0.12	≤ 100	> 3600	0.12	≤ 100	> 3600	0.12
	40	0.20	7	0	0.00	7	0	0.00	≤ 95	> 3600	0.09	≤ 95	> 3600	0.09	≤ 95	> 3600	0.09
	45	0.24	5	0	0.00	5	0	0.00	≤ 95	> 3600	0.09	≤ 95	> 3600	0.09	≤ 95	> 3600	0.09
	50	0.29	4	0	0.00	4	0	0.00	≤ 98	> 3600	0.11	≤ 98	> 3600	0.11	≤ 98	> 3600	0.11
	55	0.33	4	0	0.00	4	0	0.00	≤ 97	> 3600	0.09	≤ 97	> 3600	0.09	≤ 97	> 3600	0.09
	60	0.38	4	0	0.00	4	0	0.00	≤ 97	> 3600	0.08	≤ 97	> 3600	0.08	≤ 97	> 3600	0.08
160×160	30	0.09	13	0	0.00	13	0	0.00	≤ 102	> 3600	0.14	≤ 102	> 3600	0.14	≤ 101	> 3600	0.13
	35	0.12	10	0	0.00	10	0	0.00	≤ 99	> 3600	0.12	≤ 99	> 3600	0.12	≤ 99	> 3600	0.12
	40	0.16	8	0	0.00	8	0	0.00	≤ 100	> 3600	0.12	≤ 100	> 3600	0.12	≤ 100	> 3600	0.12
	45	0.19	7	0	0.00	7	0	0.00	≤ 95	> 3600	0.09	≤ 95	> 3600	0.09	≤ 95	> 3600	0.09
	50	0.23	5	0	0.00	5	0	0.00	≤ 94	> 3600	0.08	≤ 94	> 3600	0.08	≤ 94	> 3600	0.08
	55	0.27	5	0	0.00	5	0	0.00	≤ 98	> 3600	0.11	≤ 98	> 3600	0.11	≤ 98	> 3600	0.11
	60	0.31	4	0	0.00	4	0	0.00	≤ 97	> 3600	0.09	≤ 97	> 3600	0.09	≤ 97	> 3600	0.09

Table 5.2: Test results on random unit disk graphs with $n = 800$.

			MDS			MSDS			MGSDS			MMSDS			MGDS		
dimensions	radius	$\rho(G)$	*opt*	time	gap	*opt*	time	gap	*opt*	time	gap	*opt*	time	gap	*opt*	time	gap
120×120	50	0.36	4	0	0.00	4	0	0.00	≤ 153	> 3600	0.07	≤ 153	> 3600	0.07	≤ 153	> 3600	0.07
	55	0.42	4	0	0.00	4	0	0.00	≤ 153	> 3600	0.06	≤ 153	> 3600	0.06	≤ 153	> 3600	0.06
	60	0.47	3	1	0.00	3	0	0.00	≤ 153	> 3600	0.07	≤ 153	> 3600	0.07	≤ 153	> 3600	0.07
	65	0.53	3	0	0.00	3	1	0.00	≤ 145	> 3600	0.01	≤ 145	> 3600	0.01	≤ 145	> 3600	0.01
	70	0.59	2	1	0.00	2	0	0.00	152	1	0.00	152	1	0.00	152	1	0.00
	75	0.64	2	0	0.00	2	0	0.00	155	1	0.00	155	1	0.00	155	1	0.00
	80	0.70	2	0	0.00	2	0	0.00	158	2	0.00	158	2	0.00	158	2	0.00
140×140	50	0.28	4	0	0.00	4	0	0.00	≤ 157	> 3600	0.10	≤ 157	> 3600	0.10	≤ 157	> 3600	0.10
	55	0.33	4	0	0.00	4	0	0.00	≤ 156	> 3600	0.10	≤ 156	> 3600	0.10	≤ 156	> 3600	0.10
	60	0.38	4	0	0.00	4	0	0.00	≤ 152	> 3600	0.06	≤ 152	> 3600	0.06	≤ 152	> 3600	0.06
	65	0.43	4	0	0.00	4	0	0.00	≤ 153	> 3600	0.06	≤ 153	> 3600	0.06	≤ 153	> 3600	0.06
	70	0.47	3	1	0.00	3	0	0.00	≤ 153	> 3600	0.07	≤ 153	> 3600	0.07	≤ 153	> 3600	0.07
	75	0.52	3	1	0.00	3	0	0.00	≤ 146	> 3600	0.02	≤ 146	> 3600	0.02	≤ 146	> 3600	0.02
	80	0.57	2	1	0.00	2	0	0.00	150	1	0.00	150	1	0.00	150	1	0.00
160×160	50	0.23	6	0	0.00	6	1	0.00	≤ 148	> 3600	0.07	≤ 148	> 3600	0.07	≤ 148	> 3600	0.07
	55	0.27	5	0	0.00	5	1	0.00	≤ 155	> 3600	0.09	≤ 155	> 3600	0.09	≤ 155	> 3600	0.09
	60	0.31	4	1	0.00	4	1	0.00	≤ 157	> 3600	0.10	≤ 157	> 3600	0.10	≤ 157	> 3600	0.10
	65	0.35	4	0	0.00	4	0	0.00	≤ 155	> 3600	0.09	≤ 155	> 3600	0.09	≤ 155	> 3600	0.09
	70	0.39	4	0	0.00	4	0	0.00	≤ 153	> 3600	0.06	≤ 153	> 3600	0.06	≤ 153	> 3600	0.06
	75	0.43	4	0	0.00	4	0	0.00	≤ 153	> 3600	0.06	≤ 153	> 3600	0.06	≤ 153	> 3600	0.06
	80	0.47	3	0	0.00	3	0	0.00	≤ 153	> 3600	0.07	≤ 153	> 3600	0.07	≤ 153	> 3600	0.07
180×180	50	0.19	7	0	0.00	7	1	0.00	≤ 153	> 3600	0.10	≤ 153	> 3600	0.10	≤ 153	> 3600	0.10
	55	0.22	6	0	0.00	6	0	0.00	≤ 147	> 3600	0.06	≤ 147	> 3600	0.06	≤ 147	> 3600	0.06
	60	0.25	5	1	0.00	5	0	0.00	≤ 152	> 3600	0.07	≤ 152	> 3600	0.07	≤ 152	> 3600	0.07
	65	0.29	4	1	0.00	4	0	0.00	≤ 157	> 3600	0.11	≤ 157	> 3600	0.11	≤ 157	> 3600	0.11
	70	0.33	4	0	0.00	4	0	0.00	≤ 155	> 3600	0.09	≤ 155	> 3600	0.09	≤ 155	> 3600	0.09
	75	0.36	4	0	0.00	4	0	0.00	≤ 153	> 3600	0.07	≤ 153	> 3600	0.07	≤ 153	> 3600	0.07
	80	0.40	4	0	0.00	4	0	0.00	≤ 152	> 3600	0.05	≤ 152	> 3600	0.05	≤ 152	> 3600	0.05

Table 5.3: Test results of GR, SA and CPLEX 12.1® on Sanchis instances with $r_i = \lceil 0.3d(i) \rceil \ \forall i \in V$.

			GR				SA				CPLEX		
dimensions	range	n	best	avg	std	time	best	avg	std	time	opt	time	gap
400 × 400	60	80	27	27.6	0.5	0	26	26	0	0.1	26	0	0
	70	80	26	27.8	0.9	0	24	24	0	0.2	24	0.1	0
	80	80	25	26.5	0.8	0	23	23	0	0.2	23	0.2	0
	90	80	24	25.4	0.8	0	23	23	0	0.2	23	0.6	0
	100	80	26	26.4	0.5	0	23	23	0	0.2	23	0.9	0
	110	80	27	27.9	0.4	0	24	24	0	0.2	24	8.3	0
	120	80	26	26.9	0.8	0	23	23	0	0.3	23	2	0
600 × 600	80	100	32	33.5	0.9	0	31	31	0	0.2	31	0	0
	90	100	32	33.9	1.1	0	29	29	0	0.2	29	0	0
	100	100	31	33	1.4	0	28	28	0	0.2	28	0.1	0
	110	100	32	32.8	0.8	0	29	29	0	0.2	29	1.1	0
	120	100	31	32.9	0.9	0	28	28	0	0.2	28	0.4	0
700 × 700	70	200	66	68.1	1.3	0	60	60.3	0.4	0.5	60	0.8	0
	80	200	63	65.8	1.1	0	59	59.9	0.4	0.6	59	14.2	0
	90	200	64	67	1.5	0	58	58.8	0.4	0.7	58	314.5	0
	100	200	64	65.9	1.4	0	57	57.7	0.5	0.8	57	3165.1	5.1
	110	200	63	66.1	1.4	0	57	57	0	0.9	≤ 57	> 3600	6
	120	200	64	65.8	1.2	0	57	57.9	0.3	1	≤ 57	> 3600	0
1000 × 1000	100	200	66	68.1	1.3	0	60	60.3	0.4	0.5	60	0.5	0
	110	200	64	67.2	1.4	0	59	59.4	0.5	0.6	59	4.9	0
	120	200	64	66.3	1.3	0	58	58	0	0.7	58	301	0
	130	200	65	67	1.3	0	**58**	59	0.2	0.7	≤ 59	> 3600	0
	140	200	63	64.8	1.4	0	57	57.1	0.2	0.8	57	2429.8	0
	150	200	64	67.3	1.5	0	58	58	0	0.8	≤ 58	> 3600	4.1
	160	200	62	65.7	1.5	0	57	57.4	0.5	0.9	≤ 57	> 3600	3.7
1500 × 1500	130	250	82	83.9	1.1	0	76	76	0	0.8	76	1.3	0
	140	250	79	82	1.7	0	73	73	0	0.8	73	1.7	0
	150	250	80	82.3	1.2	0	73	73.8	0.4	0.8	73	69.6	0
	160	250	81	83.4	1	0	72	72.4	0.5	0.9	72	123.3	0
2000 × 2000	180	300	96	98	1.2	0	86	86	0	1.2	86	21.7	0
	190	300	96	98.1	1.1	0	87	87	0	1.1	≤ 87	> 3600	2.1
	200	300	98	100.9	1.6	0	87	87.2	0.4	1.2	≤ 87	> 3600	3.1
	210	300	95	97.3	1.4	0	**86**	87.2	0.6	1.2	≤ 87	> 3600	5.3
	220	300	99	100.1	0.9	0	87	87.9	0.5	1.3	≤ 87	> 3600	5.5
2500 × 2500	200	350	113	116.4	1.9	0	101	101.9	0.3	1.3	101	64.4	0
	210	350	114	116.1	1.5	0	103	103	0	1.3	$\leq$ **102**	> 3600	0
	220	350	112	114.8	1.7	0	100	100.9	0.3	1.3	100	2308.7	0
	230	350	113	115.7	1.6	0	101	102.2	0.7	1.4	≤ 101	> 3600	3.3
3000 × 3000	210	400	128	131.5	2.4	0	120	120.1	0.2	1.3	120	1.6	0
	220	400	132	136.7	2.2	0	121	121	0	1.4	121	7.3	0
	230	400	130	133.7	1.5	0	119	119.3	0.5	1.4	$\leq$ **118**	2831.5	0
	240	400	129	132.5	1.8	0	116	117	0.3	1.5	≤ 116	> 3600	2.8

Table 5.4: Test results of GR, SA and CPLEX 12.1® on Sanchis instances with twice more number of vertices with $r_i = \lceil 0.3d(i) \rceil \ \forall i \in V$. Better solution is highlighted in bold.

			GR				SA				CPLEX		
dimensions	range	n	best	avg	std	time	best	avg	std	time	opt	time	gap
400 × 400	60	160	52	54.4	1.1	0	47	47	0	0.6	47	108	0
	70	160	53	54.2	0.7	0	47	47.6	0.5	0.7	47	1497.8	0
	80	160	51	53.5	1.2	0	46	46.5	0.5	0.8	≤ 46	> 3600	4.1
	90	160	48	50.7	1.1	0	45	45.7	0.5	1	≤ 45	> 3600	4.7
	100	160	48	51	1.2	0	46	46.6	0.5	1	≤ 45	> 3600	4.2
	110	160	49	50.3	0.8	0	46	46.5	0.5	1.3	≤ 46	> 3600	6.7
	120	160	49	51	1.1	0	46	46.2	0.4	1.5	≤ 45	> 3600	4.2
600 × 600	80	200	64	66.2	1.5	0	58	58.2	0.4	0.8	58	2404.1	0
	90	200	64	67.3	1.5	0	57	58	0	0.8	≤ 57	> 3600	4.1
	100	200	64	66	1.1	0	56	56.9	0.4	1	≤ 56	> 3600	5.2
	110	200	64	66	1	0	**57**	58.4	0.6	1.1	≤ 58	> 3600	8.8
	120	200	64	65.5	1	0	58	58.3	0.4	1.2	≤ 58	> 3600	9.6
700 × 700	70	400	130	132.5	1.7	0	114	115.5	0.7	1.9	≤ 114	> 3600	7.2
	80	400	125	128.1	1.8	0	**113**	113.8	0.5	2.2	≤ 114	> 3600	10.1
	90	400	128	131.2	1.8	0	**116**	117.9	0.5	2.5	≤ 118	> 3600	13.4
	100	400	125	128.2	1.8	0	**114**	116	0.7	2.8	≤ 115	> 3600	13.6
	110	400	124	127.8	1.4	0	116	116.5	0.6	3.1	≤ 116	> 3600	14.6
	120	400	125	128.9	1.5	0	**115**	117.6	1	3.6	≤ 117	> 3600	15.5
1000 × 1000	100	400	130	132.5	1.7	0	114	115.5	0.7	1.6	≤ 114	> 3600	7.2
	110	400	126	129.3	2.2	0.1	**113**	115.4	0.6	1.8	≤ 114	> 3600	9
	120	400	127	130.9	1.7	0	**115**	116.2	0.5	2	≤ 116	> 3600	12.1
	130	400	127	130.7	2.2	0	116	117.3	0.8	2.2	≤ 116	> 3600	12.4
	140	400	125	128.1	1.4	0.1	**114**	116.4	0.5	2.5	≤ 116	> 3600	13.9
	150	400	129	130.6	1.2	0	**115**	116.3	0.8	2.6	≤ 116	> 3600	14.2
	160	400	126	128.1	1.4	0	**115**	116.6	0.9	3	≤ 116	> 3600	14.4
1500 × 1500	130	500	157	162.4	2.3	0	144	144.9	0.7	2.7	≤ 144	> 3600	8.9
	140	500	159	163.7	2.5	0	**143**	145.1	0.7	2.7	≤ 146	> 3600	11.6
	150	500	162	164.9	1.8	0	**145**	146.9	0.7	2.9	≤ 147	> 3600	12.5
	160	500	161	163.9	1.6	0	**144**	146.2	0.9	2.8	≤ 147	> 3600	13.3
2000 × 2000	180	600	193	197.3	2.2	0	**172**	173.8	0.9	3.4	≤ 173	> 3600	11.9
	190	600	194	197.3	2.1	0	**175**	176	0.9	3.8	≤ 176	> 3600	13.2
	200	600	195	199	2.6	0	**173**	176	0.8	3.8	≤ 176	> 3600	14.2
	210	600	193	196.5	2.4	0	**174**	175.8	0.6	4	≤ 176	> 3600	14.9
	220	600	189	193.4	1.8	0	173	174.7	0.8	4.2	$\leq$ **172**	> 3600	13.7
2500 × 2500	200	700	223	227.3	1.9	0	**201**	202.9	0.9	4.1	≤ 203	> 3600	12.6
	210	700	224	227.6	2.4	0	**200**	201.6	0.8	4.5	≤ 202	> 3600	12.8
	220	700	225	229	2.3	0	**201**	204.1	1	4.5	≤ 206	> 3600	14.2
	230	700	224	228.5	1.9	0	**203**	205	0.7	4.6	≤ 205	> 3600	14.5
3000 × 3000	210	800	259	265.3	2.9	0	**229**	230.8	0.9	4.6	≤ 234	> 3600	12.1
	220	800	256	263.2	2.8	0.1	**229**	231.3	0.7	5	≤ 234	> 3600	13.1
	230	800	253	259	2.6	0	**228**	230.5	0.7	5.1	≤ 230	> 3600	12.2
	240	800	256	259.9	2.3	0	**229**	231	1	5.3	≤ 233	> 3600	14.2

Table 5.5: Test results of GR, SA and CPLEX 12.1® on Sanchis instances with three times more number of vertices with $r_i = \lceil 0.3d(i) \rceil \ \forall i \in V$. Better solution is highlighted in bold.

			GR				SA				CPLEX		
dimensions	range	n	best	avg	std	time	best	avg	std	time	opt	time	gap
400 × 400	60	240	78	79.9	1.3	0	69	69.3	0.4	0.8	≤ 69	> 3600	7.3
	70	240	78	79.1	0.9	0.1	70	70.4	0.5	0.9	≤ 70	> 3600	11.8
	80	240	77	78.2	0.8	0	69	70.1	0.7	1.2	≤ 69	> 3600	11.1
	90	240	76	78.4	1.1	0	69	70.4	0.7	1.6	≤ 69	> 3600	12.1
	100	240	75	78.8	1.6	0	70	71.2	0.6	1.9	≤ 70	> 3600	13.1
	110	240	75	76.8	0.9	0	70	70.7	0.5	2.4	≤ **69**	> 3600	12.2
	120	240	75	76.5	1.1	0	69	69.5	0.5	3.2	≤ **68**	> 3600	10.9
600 × 600	80	300	96	98.9	1.6	0	**85**	86.1	0.6	1.1	≤ 86	> 3600	9.3
	90	300	95	96.9	1.2	0	**85**	86.5	0.7	1.4	≤ 87	> 3600	12.1
	100	300	94	96.8	1.1	0	**86**	87.2	0.5	1.6	≤ 88	> 3600	13.5
	110	300	94	96.5	1.3	0	88	88.5	0.5	1.7	≤ 88	> 3600	14.3
	120	300	94	96.1	1.1	0	**87**	88.2	0.6	2.2	≤ 89	> 3600	15.5
700 × 700	70	600	194	198.7	2.4	0	**173**	175.5	1	3.7	≤ 177	> 3600	14.8
	80	600	188	192.8	2.2	0	**174**	175.3	0.9	4.2	≤ 177	> 3600	16.3
	90	600	194	196.7	1.5	0.1	**176**	177.2	0.8	5.3	≤ 179	> 3600	17.8
	100	600	192	194.8	1.4	0	**177**	179	1.1	6.4	≤ 178	> 3600	17.7
	110	600	193	195.6	1.4	0	177	178.2	0.8	7.7	≤ **176**	> 3600	17
	120	600	196	199.9	2	0.1	**180**	181	0.7	9.4	≤ 181	> 3600	19.3
1000 × 1000	100	600	194	198.7	2.4	0	**173**	175.5	1	3.7	≤ 174	> 3600	13.3
	110	600	189	192.7	1.9	0	173	174.5	0.7	4.2	≤ 173	> 3600	14.3
	120	600	191	193.7	1.5	0	**174**	175.6	0.9	5	≤ 177	> 3600	16.4
	130	600	194	196	1.2	0	**177**	177.9	0.8	5.3	≤ 180	> 3600	18.2
	140	600	192	196.2	1.7	0	**177**	178.8	0.8	6	≤ 179	> 3600	18.1
	150	600	191	193.6	1.7	0	**176**	178.7	1	6.6	≤ 181	> 3600	18.9
	160	600	191	194.6	2.2	0.1	**176**	178.5	0.9	8	≤ 177	> 3600	17.5
1500 × 1500	130	750	240	244.3	2.5	0	**218**	219.3	0.8	4.9	≤ 221	> 3600	14.9
	140	750	239	242	1.8	0	**217**	219.7	1.1	5.3	≤ 219	> 3600	15.3
	150	750	243	246.1	2.1	0.1	**221**	222.2	1	5.8	≤ 223	> 3600	17
	160	750	237	241.8	2.2	0	**218**	220.4	1.1	6.4	≤ 223	> 3600	17.6
2000 × 2000	180	900	286	290.7	2	0.1	**262**	263.5	0.9	7.3	≤ 266	> 3600	16.9
	190	900	287	291.4	2.1	0.1	**264**	266	1.2	7.6	≤ 272	> 3600	19
	200	900	291	295.7	2.9	0	**267**	268.7	0.9	8.4	≤ 270	> 3600	18.3
	210	900	288	291.4	2	0.1	**265**	267.8	1.2	9.4	≤ 268	> 3600	17.8
	220	900	286	289.2	2.1	0	**267**	268.3	0.7	9.6	≤ 270	> 3600	18.9
2500 × 2500	200	1050	333	338.2	2.5	0	**305**	307	1	8.9	≤ 313	> 3600	17.9
	210	1050	336	340.1	2.7	0	**305**	306.7	1.2	9.6	≤ 310	> 3600	17.6
	220	1050	336	339.7	2	0	**309**	311.3	1	10	≤ 320	> 3600	20
	230	1050	335	338.8	2.5	0	**308**	310.7	1	10.5	≤ 314	> 3600	18.7
3000 × 3000	210	1200	387	392.4	2.6	0	**349**	352	1.5	12.4	≤ 356	> 3600	16.3
	220	1200	383	389.4	2.8	0	**347**	350	1.6	12.4	≤ 353	> 3600	16.6
	230	1200	383	387.7	2.6	0	**352**	353.3	0.9	13	≤ 357	> 3600	18.1
	240	1200	381	385.9	2.6	0.1	**352**	354.6	1.4	14	≤ 361	> 3600	19.1

Table 5.6: Test results of GR, SA and CPLEX 12.1® on DIMACS instances with $r_i = \lceil 0.1d(i) \rceil \ \forall i \in V$. Better solution is highlighted in bold.

Graphs			GR				SA				CPLEX		
	n	m	best	avg	std	time	best	avg	std	time	opt	time	gap
karate	34	78	4	4	0	0	4	4	0	0.1	4	0	0
dolphins	62	159	15	15.7	0.6	0	14	14	0	0.1	14	0	0
polbooks	105	441	13	15	0.9	0	13	13	0	0.3	13	0	0
adjnoun	112	425	18	18.6	0.7	0	18	18	0	0.3	18	0	0
football	115	613	21	21.8	0.7	0	19	19.1	0.2	0.2	19	11.4	0
jazz	198	2742	20	22	1	0	18	18	0	1.7	18	0.1	0
celegans_metabolic	453	2025	32	32	0	0	30	30	0	4.9	30	0	0
email	1133	5451	230	232.8	1.6	0.1	213	213.1	0.2	11.5	213	0	0
polblogs	1490	16715	144	146.8	1.5	0.1	131	131	0	95.5	131	0.1	0
netscience	1589	2742	353	355.5	1.2	0	351	351	0	6.6	351	0	0
delaunay_n10	1024	3056	181	186.8	3.4	0	157	158	0.6	3.8	**156**	2028.5	0
delaunay_n11	2048	6127	369	375.3	3.7	0.1	317	318.3	0.9	15.2	≤ 317	> 3600	3.1
delaunay_n12	4096	12264	750	757.5	5.2	0.1	629	631.3	0.9	63.7	≤ 629	> 3600	3.2

Table 5.7: Test results of GR, SA and CPLEX 12.1® on DIMACS instances with $r_i = \lceil 0.3d(i) \rceil \ \forall i \in V$. Better solution is highlighted in bold.

Graphs			GR				SA				CPLEX		
	n	m	best	avg	std	time	best	avg	std	time	opt	time	gap
karate	34	78	7	7	0	0	6	6	0	0	6	0	0
dolphins	62	159	19	20.4	0.7	0	17	17	0	0.1	17	0	0
polbooks	105	441	27	28.5	0.7	0	25	25	0	0.3	25	0.1	0
adjnoun	112	425	27	28.5	0.6	0	25	25.1	0.2	0.2	25	0	0
football	115	613	38	39.7	0.9	0	35	35.9	0.4	0.2	≤ 35	> 3600	7.8
jazz	198	2742	44	45.3	0.7	0	42	42.3	0.5	1.6	42	5.1	0
celegans_metabolic	453	2025	65	66.4	0.7	0	56	56	0	3.6	56	0	0
email	1133	5451	302	306.4	2.4	0	269	270	0.3	11.7	≤ 269	> 3600	0.7
polblogs	1490	16715	208	212.1	1.7	0.1	175	176	0.2	88.1	175	160.9	0
netscience	1589	2742	438	440.3	1.5	0	433	433	0	6.5	433	0.1	0
delaunay_n10	1024	3056	330	338.2	3.8	0	287	288.3	0.8	4	≤ 287	> 3600	2.7
delaunay_n11	2048	6127	661	678.2	6.5	0.1	**576**	576.9	0.8	15.5	≤ 577	> 3600	3.3
delaunay_n12	4096	12264	1346	1359.1	7.3	0.1	**1150**	1153.1	1.4	65.1	≤ 1156	> 3600	3.6

Table 5.8: Test results of GR, SA and CPLEX 12.1® on DIMACS instances with $r_i = \lceil 0.5d(i) \rceil \; \forall i \in V$. Better solution is highlighted in bold.

Graphs			GR				SA				CPLEX		
	n	m	best	avg	std	time	best	avg	std	time	opt	time	gap
karate	34	78	9	11.3	0.9	0	9	9	0	0	9	0	0
dolphins	62	159	22	23.6	0.8	0	22	22	0	0.1	22	0	0
polbooks	105	441	36	39	1.4	0	34	34	0	0.2	34	0.8	0
adjnoun	112	425	37	37.9	0.7	0	34	34.3	0.4	0.2	34	0.1	0
football	115	613	54	56.3	0.8	0	51	51.1	0.2	0.2	≤ 51	> 3600	10.2
jazz	198	2742	72	74	1.4	0	67	67.6	0.5	1.2	67	2422.9	0
celegans_metabolic	453	2025	89	91.3	1.1	0	85	85	0	3.2	85	0.1	0
email	1133	5451	384	389	2.8	0	**345**	345.8	0.6	10.5	≤ 346	> 3600	3.9
polblogs	1490	16715	292	297.7	3	0.2	246	247.2	0.5	68.2	≤ 246	> 3600	3.8
netscience	1589	2742	540	542	1.2	0	533	533	0	6.4	533	0.6	0
delaunay_n10	1024	3056	436	442.8	3.7	0	388	389.4	0.9	3.8	≤ 388	> 3600	8.2
delaunay_n11	2048	6127	868	880	4.1	0.1	**762**	763.8	0.8	15.4	≤ 767	> 3600	8.4
delaunay_n12	4096	12264	1745	1765.8	8.3	0.2	**1546**	1549	1.6	64.5	≤ 1565	> 3600	10.3

Table 5.9: Test results of GR, SA and CPLEX 12.1® on DIMACS instances with $r_i = \lceil 0.7d(i) \rceil \; \forall i \in V$. Better solution is highlighted in bold.

Graphs			GR				SA				CPLEX		
	n	m	best	avg	std	time	best	avg	std	time	opt	time	gap
karate	34	78	13	13.8	0.7	0	13	13	0	0	13	0.1	0
dolphins	62	159	31	31.9	0.9	0	29	29	0	0	29	0.6	0
polbooks	105	441	52	53.9	1.5	0	50	50	0	0.2	50	0	0
adjnoun	112	425	54	56.3	1.1	0	50	50	0	0.2	50	0	0
football	115	613	69	72	1.1	0	66	66.5	0.5	0.2	66	3.7	11.4
jazz	198	2742	106	107.9	0.9	0	101	101.6	0.5	0.8	≤ 101	> 3600	5.2
celegans_metabolic	453	2025	200	203.3	2.1	0	188	188.8	0.4	1.7	188	0.6	0
email	1133	5451	541	544.8	2.3	0	490	490.6	0.6	8	≤ 490	> 3600	2.6
polblogs	1490	16715	435	443.1	3.8	0.2	**378**	378.9	0.6	43.6	≤ 380	> 3600	4.4
netscience	1589	2742	804	806.4	1.3	0.1	803	803	0	6.1	803	0.8	0
delaunay_n10	1024	3056	628	633.5	3.1	0.1	**572**	574.2	0.8	3.6	≤ 577	> 3600	19.7
delaunay_n11	2048	6127	1248	1257	6.5	0.1	**1137**	1139.5	1.3	14.5	≤ 1150	> 3600	20.6
delaunay_n12	4096	12264	2515	2526	6.7	0.2	**2281**	2284.3	1.7	63.1	≤ 2308	> 3600	21.2

Table 5.10: Test results of GR, SA and CPLEX 12.1® on DIMACS instances with $r_i = \lceil 0.9d(i) \rceil\ \forall i \in V$. Better solution is highlighted in bold.

Graphs			GR				SA				CPLEX		
	n	m	best	avg	std	time	best	avg	std	time	opt	time	gap
karate	34	78	14	14.3	0.4	0	14	14	0	0	14	0	0
dolphins	62	159	34	35.8	0.9	0	34	34	0	0.1	34	0	0
polbooks	105	441	63	65.3	1.6	0	62	62	0	0.2	62	0	0
adjnoun	112	425	62	64.3	1.3	0	58	58.1	0.2	0.2	58	0.1	0
football	115	613	87	88.9	1.4	0	83	83.1	0.3	0.2	≤ 83	> 3600	10.2
jazz	198	2742	145	147.7	1.4	0.1	140	140	0	0.5	≤ 140	> 3600	9.4
celegans_metabolic	453	2025	263	265.7	1.5	0	248	248	0	1.2	248	0.1	0
email	1133	5451	616	622.7	3.4	0	578	578	0	6.3	578	67.5	0
polblogs	1490	16715	546	554.3	5.4	0.2	505	505	0	28.2	$\leq$ **504**	> 3600	1.3
netscience	1589	2742	898	898.7	0.5	0	898	898	0	5.6	898	0.1	0
delaunay_n10	1024	3056	754	759.4	3.1	0	703	703.8	0.7	3.2	703	174.7	0
delaunay_n11	2048	6127	1504	1512.7	6.5	0.1	1402	1402.8	0.8	13	≤ 1402	> 3600	0.3
delaunay_n12	4096	12264	3023	3035.9	7.8	0.2	2809	2811.9	1.1	56.5	$\leq$ **2808**	> 3600	0.4

Table 5.11: Test results of GR and SA on large DIMACS instances with $r_i = \lceil 0.2d(i) \rceil \ \forall i \in V$.

Graphs			GR				SA			
	n	m	best	avg	std	time	best	avg	std	time
Clustering										
hep-th	8361	15751	1957	1961.3	3.7	0.0	1904	1904.0	0.0	23.7
PGPgiantcompo	10680	24316	2829	2830.3	1.3	0.7	2751	2751.0	0.0	49.7
cond-mat	16726	47594	3346	3347.0	0.8	0.7	3223	3223.0	0.0	74.3
smallworld	100000	499998	24416	24444.7	22.9	35.0	22509	22531.7	17.3	1138.3
G_n_pin_pout	100000	501198	24422	24437.3	11.6	36.3	23236	23258.3	18.0	1537.0
Walshaw										
3elt	4720	13722	1400	1404.3	5.4	0.0	1148	1151.0	2.5	15.7
uk	4824	6837	1583	1589.7	4.8	0.0	1379	1382.3	4.0	14.0
add32	4960	9462	1222	1222.0	0.0	0.0	1222	1222.0	0.0	14.0
whitaker3	9800	28989	2989	2991.3	3.3	0.3	2440	2444.3	4.8	35.0
crack	10240	30380	2302	2322.0	15.6	0.0	1952	1957.3	3.8	51.7
wing_nodal	10937	75488	2533	2535.3	2.1	0.3	2269	2272.3	4.0	66.7
fe_4elt2	11143	32818	3128	3149.3	20.0	0.3	2596	2598.7	1.9	51.3
4elt	15606	45878	4721	4736.7	12.0	1.3	3882	3883.3	0.9	73.3
fe_sphere	16386	49152	5185	5205.0	15.0	1.3	4226	4230.0	3.3	82.0
cti	16840	48232	5157	5169.3	10.7	1.7	4693	4694.0	0.8	109.7
memplus	17758	54196	2520	2520.7	0.5	1.0	2487	2494.7	5.8	121.7
cs4	22499	43858	6179	6190.3	8.4	2.0	5370	5379.7	8.2	117.0
fe_pwt	36519	144794	9497	9510.3	10.6	5.0	8170	8173.0	4.2	208.3
fe_body	45087	163734	10185	10199.7	11.5	7.0	8864	8868.7	4.6	266.0
t60k	60005	89440	18149	18185.7	45.0	18.0	16109	16147.3	27.6	330.3
wing	62032	121544	16966	16983.0	12.0	17.0	14968	14980.7	11.6	380.0
Delaunay										
delaunay_n13	8192	24547	2053	2061.3	6.9	1.0	1757	1759.3	2.1	28.0
delaunay_n14	16384	49122	4109	4119.3	7.3	1.0	3535	3539.0	3.7	80.7
delaunay_n15	32768	98274	8207	8222.0	10.7	4.0	7083	7088.3	5.0	179.0
delaunay_n16	65536	196575	16454	16461.7	9.5	15.7	14290	14297.0	6.7	397.7
delaunay_n17	131072	393176	32889	32909.3	23.3	67.0	28795	28806.3	8.2	1070.0
delaunay_n18	262144	786396	65780	65887.0	82.9	285.0	58116	58137.7	17.2	2093.3
Rand										
rgg_n_2_15_s0	32768	160240	7986	8004.0	25.5	3.7	7029	7035.0	7.8	175.7
rgg_n_2_16_s0	65536	342127	15863	15894.7	22.5	14.7	14195	14197.3	2.6	388.0
rgg_n_2_17_s0	131072	728753	31647	31660.3	18.2	64.3	28562	28570.7	7.9	1049.7
rgg_n_2_18_s0	262144	1547283	62792	62816.7	20.4	283.7	57546	57571.0	23.3	2201.3

Table 5.12: Test results of GR and SA on large DIMACS instances with $r_i = \lceil 0.3d(i) \rceil \ \forall i \in V$.

Graphs			GR				SA			
	n	m	best	avg	std	time	best	avg	std	time
Clustering										
hep-th	8361	15751	2158	2164.3	4.6	0.0	2070	2070.7	0.5	24.7
PGPgiantcompo	10680	24316	2974	2982.7	6.3	0.7	2869	2869.0	0.0	49.0
cond-mat	16726	47594	4000	4003.3	2.9	1.0	3836	3837.0	0.8	73.7
smallworld	100000	499998	32570	32602.7	40.0	48.3	30250	30285.0	34.5	1127.0
G_n_pin_pout	100000	501198	32221	32244.3	25.6	47.3	30459	30492.3	35.8	1473.7
Walshaw										
3elt	4720	13722	1556	1562.0	4.6	0.3	1283	1285.3	2.6	15.7
uk	4824	6837	1583	1589.7	4.8	0.3	1379	1382.3	4.0	14.0
add32	4960	9462	1222	1222.7	0.5	0.3	1222	1222.0	0.0	13.7
whitaker3	9800	28989	3159	3168.0	8.3	0.7	2617	2618.0	0.8	34.0
crack	10240	30380	3141	3151.7	8.2	0.3	2707	2711.7	3.4	52.0
wing_nodal	10937	75488	3461	3471.0	7.1	1.0	3163	3169.0	4.6	68.3
fe_4elt2	11143	32818	3657	3667.0	11.4	0.7	3047	3048.7	1.7	51.0
4elt	15606	45878	5102	5117.3	12.7	1.3	4229	4233.3	4.8	74.7
fe_sphere	16386	49152	5139	5163.7	24.5	1.7	4229	4232.3	3.4	82.3
cti	16840	48232	5857	5868.0	14.9	1.3	5314	5328.0	10.7	108.3
memplus	17758	54196	2571	2571.0	0.0	1.0	2538	2538.0	0.0	131.3
cs4	22499	43858	9573	9579.0	7.8	3.3	8303	8316.0	11.5	117.7
fe_pwt	36519	144794	13217	13230.0	9.6	7.3	12106	12108.7	2.5	205.7
fe_body	45087	163734	14454	14475.0	24.3	9.0	13001	13006.7	8.0	275.3
t60k	60005	89440	18149	18185.7	45.0	18.0	16109	16147.3	27.6	326.7
wing	62032	121544	26549	26568.0	13.5	26.0	23263	23274.7	9.4	377.7
Delaunay										
delaunay_n13	8192	24547	2689	2705.3	13.9	0.0	2313	2316.0	2.2	29.3
delaunay_n14	16384	49122	5397	5401.0	2.9	1.3	4640	4643.0	2.5	82.0
delaunay_n15	32768	98274	10778	10781.3	2.9	5.0	9320	9327.7	6.6	180.0
delaunay_n16	65536	196575	21580	21603.0	25.8	20.7	18762	18769.7	6.6	396.7
delaunay_n17	131072	393176	43144	43198.3	48.6	88.7	37802	37806.7	5.3	1062.7
delaunay_n18	262144	786396	86132	86187.3	44.9	386.7	76148	76162.3	10.1	2078.0
Rand										
rgg_n_2_15_s0	32768	160240	10803	10811.3	5.9	5.3	9694	9702.7	7.4	183.0
rgg_n_2_16_s0	65536	342127	21562	21572.3	8.6	20.7	19542	19549.0	5.4	403.0
rgg_n_2_17_s0	131072	728753	42835	42852.0	15.3	90.0	39326	39341.7	16.2	1070.3
rgg_n_2_18_s0	262144	1547283	85309	85328.7	17.3	381.3	79441	79453.0	13.0	2208.3

6. CONCLUSION AND FUTURE WORK

In Chapter 3, we consider the problem of finding the smallest connected dominating set with bounded diameter (i.e., dominating s-club or DsC). A DsC can be used as a reliable virtual backbone in wireless ad-hoc network. We prove that the problem of checking the existence of a DsC is NP-complete for any fixed positive integer s. A sufficient condition for a graph $G = (V, E)$ to have a DsC is provided. Solving MDsC problem is shown to be NP-hard even when it is restricted to graphs for which a dominating clique is known to exist. The first exact approach to solve minimum DsC problem with $O(sn^2)$ entities is proposed with some numerical experiment results using CPLEX on random unit disk graphs. We investigate the characteristics of a graph G having a DsC, and develop some valid inequalities and variable fixing techniques for this problem.

If we generalize MDsC problem such that s depends on some input such as the number of vertices in the graph $G = (V, E)$, then we can solve MCDS problem by solving MDsC problem. For a naive approach, simply let $s = n - 1$, where $|V| = n$. More practically, if there is an upper bound U for $\gamma_c(G)$, a solution to MDsC problem with $s = (U - 1)$ will solve MCDS problem. We might use a MCDS heuristic to find a good upper bound U on $\gamma_c(G)$ and solve MDsC problem with $s = U - 1$ to find an optimal solution to MCDS problem. We can also easily show that if $P \subset V$ solves the MDsC problem in G and $|P| \leq s + 2$, then P also solves the MCDS problem in G. Thus, we can solve a series of MDsC problems for different values of s to solve MCDS problem. Start with $s = diam(G) - 2$ and solve the MDsC problem. If the solution also solves MCDS problem, we are done. Otherwise, increase s by 1 and repeat. The preliminary computational experiment results using the formulation for

MDsC problem to solve MCDS problem reveal no better performance compared to other methods for MCDS problem introduced in [78, 34], especially on instances with large diameter. However, on instances with a small CDS, MDsC formulation was shown to be competitive. We leave more comprehensive examination of the fitness of the proposed approach for the MCDS problem for future work. Another interesting direction of future research could be about creating a DsC which is robust under edge and vertex failures. A robust structure called R-robust s-club is introduced in [98]. Not only it remains connected after the failure of $R-1$ vertices, but it also maintains diameter s. For a robust version of DsC, an interesting structure to study may be an R-robust dominating s-club, a structure that maintains both dominating nature and diameter s after the failure of $R-1$ dominating vertices.

Note that the domination constraints (3.4) of DsC (or CDS) could be replaced by the following constraints. Let D be a DsC (or CDS). For each vertex $i \in V$, let $y_i = |D \cap N(i)|$. Assume there is no vertex dominating the entire graph. Then we have $1 \leq y_i \leq |N(i)|$ for $i \in V$. If we have a lower bound p for DsC (or CDS), then p vertices must be connected and we have $\sum_{i \in V} y_i \geq 2(p-1)$. The domination constraints (3.4) could be replaced by the following set of constraints:

$$\sum_{j \in N(i)} x_j \geq y_i \ \forall i \in V; \tag{6.1}$$

$$\sum_{i \in V} y_i \geq 2(p-1); \tag{6.2}$$

$$1 \leq y_i \leq |N(i)|. \tag{6.3}$$

For lower bound p, since the size of the minimum total dominating set provides a lower bound for the size of minimum connected dominating set ($\gamma_t \leq \gamma_c$), and also it is much easier to solve than MCDS problem, we can use the following value:

$$p = \min\{\sum_{i \in V} x_i | \sum_{j \in N(i)} x_j \geq 1, i \in V, x \in \{0,1\}^n\}.$$

Moreover, since $n - p$ vertices which are not in the minimum total dominating set must also be adjacent to at least one vertex in D, the constraint (6.2) can be modified to $\sum_{i \in V} y_i \geq 2(p-1) + n - p$. The preliminary experiments indicate that this modification could be more effective for sparse instances than the original formulation. We leave more in-depth investigation of this extension as future work.

In Chapter 4, we generalize the well-studied connected dominating set (CDS) and study the k-connected d-dominating set (k-d-CDS). To the best of my knowledge, this is the first study concerning an exact approach for minimum k-d-CDS (Mk-d-CDS) problem. We demonstrate that several structural properties hold for $d \geq k$, but fail for $d < k$. Robustness is a desirable characteristic but it is not free. Depending on the value of k and d, we have three different mathematical programming formulations. Knowing that minimal vertex-cut inequality may not necessarily induce facets, r-robust vertex-cut and r-robust vertex-cut inequalities are developed. For minimum 1-1-CDS problem (or MCDS problem), we show that the proposed solution method using lazy-constraint compares favorably with existing approaches. For $k = d = 2, 3, 4$, the comparison between minimum k-total dominating set problem and Mk-k-CDS problem demonstrates that the connectivity constraints may not be burdensome. We show the necessary and sufficient conditions for a graph $G = (V, E)$ to have a k-d-CDS when $d \geq k$. The conditions for a graph $G = (V, E)$ to have a k-d-CDS when $d < k$ is still unknown. The separation problem for the vertex-cut inequality (or 1-robust vertex-cut inequality) is weighted vertex-connectivity problem, and it can be solved in polynomial time. However, the complexity of separation for r-robust vertex-cut inequalities when $r \geq 2$ is still an open question. In terms of computational experiments, one might be interested in applying the proposed approach on different types of graphs with low/high connectivity and low/high density. A few articles, mostly from the computer science community, consider heuristic algorithms to solve Mk-d-CDS problem [29, 75, 104] and present computational results. It might be interesting to apply the proposed approach on the same instances considered in [29, 75, 104] to evaluate the heuristics used.

Chapter 5 introduces several varieties of the classical dominating set motivated by real life scenarios of wireless sensor placement problem. Computational complexities show that all of the newly introduced problems are hard in their own respect. Mathematical programming formulations and some basic polyhedral properties are presented. In particular, for MGDS problem, some analytical bounds on generalized domination number compared to domination number are presented. Numerical experiment results demonstrate that some varieties could be practically much more challenging to solve than others for some commercial solvers. For MGDS problem, simulated annealing is also applied and its performance is compared to that of CPLEX 12.1®. For almost all considered instances, simulated annealing compares favorably with CPLEX 12.1®. For future research, we might want to consider other metaheuristic algorithms such as *genetic algorithm* and *ant colony optimization* to tackle the problems.

We conclude this dissertation by introducing a new interesting dominating set related problem. We might consider the problem of finding the maximum subgraph that can be dominated by $1 \leq k(\leq \gamma(G))$ vertices, where $\gamma(G)$ is the domination number. This problem, say *maximum k-coverage* problem, is at least not easier than MDS problem, otherwise we could solve MDS problem by increasing the value of k until the entire graph is dominated in polynomial time. A mathematical programming formulation for this problem can be presented in the following way.

$$
\begin{aligned}
\text{maximize } \sum_{i=1}^{n} x_i & & & \qquad (6.4)\\
\text{s.t. } x_i & \geq & \frac{\sum_{j\in N[i]} y_j}{k},\ i \in V & \qquad (6.5)\\
x_i & \leq & \sum_{j\in N[i]} y_j,\ i \in V & \qquad (6.6)\\
\sum_{i=1}^{n} y_i & = & k & \qquad (6.7)\\
x_i, y_i & \in & \{0,1\},\ i \in \{1,2,\ldots,n\}, & \qquad (6.8)
\end{aligned}
$$

where $x_i = 1$ iff vertex i is dominated and $y_i = 1$ iff vertex i is in the dominating set. Constraint (6.5) makes a vertex i to be dominated if at least one vertex from $N[i]$ is in the dominating set, while constraint (6.6) guarantees that i is not dominated if none of $N[i]$ is in the dominating set. Constraint (6.7) restricts the number of vertices in the dominating set to be k, which is given as an input parameter. We leave more in-depth investigation about this problem for future research.

REFERENCES

[1] J. Abello, M. Resende, and S. Sudarsky. Massive quasi-clique detection. *LATIN 2002: Theoretical Informatics*, pages 598–612, 2002.

[2] J. Alber, M. Fellows, and R. Niedermeier. Polynomial-time data reduction for dominating set. *J. ACM*, 51:363–384, May 2004.

[3] M.T. Almeida and F.D. Carvalho. Integer models and upper bounds for the 3-club problem. *Networks*, 2012.

[4] K.M. Alzoubi, P.J. Wan, and O. Frieder. Distributed heuristics for connected dominating sets in wireless ad hoc networks. *Journal of Communications and Networks*, 4(1):22–29, 2002.

[5] B. Balasundaram, S. Butenko, and I. Hicks. Clique relaxations in social network analysis: The maximum k-plex problem. *Operations Research*, 59(1):133–142, 2011.

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

[7] L. Bao and J.J. Garcia-Luna-Aceves. Topology management in ad hoc networks. In *Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*, pages 129–140. ACM, 2003.

[8] J. Blum, M. Ding, A. Thaeler, and X. Cheng. Connected dominating set in sensor networks and manets. *Handbook of Combinatorial Optimization*, pages 329–369, 2005.

[9] I.M. Bomze, M. Budinich, P.M. Pardalos, and M. Pelillo. The maximum clique problem. *Handbook of Combinatorial Optimization*, 4(1):1–74, 1999.

[10] M. Bóna. *A Walk Through Combinatorics: An Introduction to Enumeration and Graph Theory*. World Scientific, 2011.

[11] K.S. Booth and J.H. Johnson. Dominating sets in chordal graphs. *SIAM Journal on Computing*, 11:191, 1982.

[12] S. Borgatti. Identifying sets of key players in a social network. *Computational & Mathematical Organization Theory*, 12(1):21–34, 2006.

[13] M. Bouchakour, TM Contenza, CW Lee, and A.R. Mahjoub. On the dominating set polytope. *European Journal of Combinatorics*, 29(3):652–661, 2008.

[14] M. Bouchakour and A.R. Mahjoub. One-node cutsets and the dominating set polytope. *Discrete Mathematics*, 165:101–123, 1997.

[15] J.M. Bourjolly, G. Laporte, and G. Pesant. Heuristics for finding k-clubs in an undirected graph. *Computers & Operations Research*, 27(6):559–569, 2000.

[16] J.M. Bourjolly, G. Laporte, and G. Pesant. An exact algorithm for the maximum k-club problem in an undirected graph. *European Journal of Operational Research*, 138(1):21–28, 2002.

[17] A. Brandstädt and D. Kratsch. On the restriction of some np-complete graph problems to permutation graphs. In *Fundamentals of Computation Theory*, pages 53–62. Springer, 1985.

[18] A. Brandstädt and D. Kratsch. On domination problems for permutation and other graphs. *Theoretical Computer Science*, 54(2-3):181–198, 1987.

[19] S. Butenko, X. Cheng, C Oliveira, and PM Pardalos. A new heuristic for the minimum connected dominating set problem on ad hoc wireless networks.

Recent Developments in Cooperative Control and Optimization, pages 61–73, 2004.

[20] Y. Caro and Y.Roditty. A note on the k-domination number of a graph. *International Journal of Math and Math Science*, 13:205–206, 1990.

[21] F.D. Carvalho and M.T. Almeida. Upper bounds and heuristics for the 2-club problem. *European Journal of Operational Research*, 210(3):489–494, 2011.

[22] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris. Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks. *Wireless Networks*, 8(5):481–494, 2002.

[23] X. Cheng, X. Huang, D. Li, W. Wu, and D.Z. Du. A polynomial-time approximation scheme for the minimum-connected dominating set in ad hoc wireless networks. *Networks*, 42(4):202–208, 2003.

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

[25] E. Cockayne, S. Goodman, and S. Hedetniemi. A linear algorithm for the domination number of a tree. *Information Processing Letters*, 4(2):41–44, 1975.

[26] D.G. Corneil and Y. Perl. Clustering and domination in perfect graphs. *Discrete Applied Mathematics*, 9(1):27–39, 1984.

[27] M.B. Cozzens and L.L. Kelleher. Dominating cliques in graphs. *Discrete Mathematics*, 86(1-3):101–116, 1990.

[28] F. Dai and J. Wu. An extended localized algorithm for connected dominating set formation in ad hoc wireless networks. *Parallel and Distributed Systems, IEEE Transactions on*, 15(10):908–920, 2004.

[29] F. Dai and J. Wu. On constructing k-connected k-dominating set in wireless ad hoc and sensor networks. *Journal of Parallel and Distributed Computing*, 66(7):947–958, 2006.

[30] B. Das, R. Sivakumar, and V. Bharghavan. Routing in ad hoc networks using a spine. In *Computer Communications and Networks, 1997. Proceedings., Sixth International Conference on*, pages 34–39. IEEE, 1997.

[31] DIMACS. Tenth dimacs implementation challenges - graph partitioning and graph clustering. *Online : http://www.cc.gatech.edu/dimacs10/index.shtml. Accessed November 2011.*, 2011.

[32] D. Du. *Connected Dominating Set: Theory and Applications*, volume 77. Springer, 2012.

[33] A. Ephremides, J.E. Wieselthier, and D.J. Baker. A design concept for reliable mobile radio networks with frequency hopping signaling. *Proceedings of the IEEE*, 75(1):56–73, 1987.

[34] N. Fan and J.P. Watson. Solving the connected dominating set problem and power dominating set problem by integer programming. *Combinatorial Optimization and Applications*, pages 371–383, 2012.

[35] M. Farber. Domination, independent domination, and duality in strongly chordal graphs. *Discrete Applied Mathematics*, 7(2):115–130, 1984.

[36] J. F. Fink and M. S. Jacobson. *On n-domination, n-dependence and forbidden subgraphs*, pages 301–311. John Wiley & Sons, Inc., New York, NY, USA, 1985.

[37] J.F. Fink and M.S. Jacobson. n-domination in graphs. In *Graph theory with applications to algorithms and computer science*, pages 283–300. John Wiley

& Sons, Inc., 1985.

[38] F. Fomin, D. Kratsch, and G. Woeginger. Exact (exponential) algorithms for the dominating set problem. In *Graph-Theoretic Concepts in Computer Science*, pages 245–256. Springer, 2005.

[39] F.V. Fomin, F. Grandoni, and D. Kratsch. Solving connected dominating set faster than 2 n. *Algorithmica*, 52(2):153–166, 2008.

[40] T. Fujie. An exact algorithm for the maximum leaf spanning tree problem. *Computers & Operations Research*, 30(13):1931–1944, 2003.

[41] T. Fujie. The maximum-leaf spanning tree problem: Formulations and facets. *Networks*, 43(4):212–223, 2004.

[42] X. Gao, B. Xu, and J. Li. A distributed design for minimum 2-connected m-dominating set in bidirectional wireless ad-hoc networks. *Tsinghua Science and Technology*, 17(5):553–566, 2012.

[43] M.R. Garey and D.S. Johnson. Computers and intractability. *A Guide to the Theory of NP-Completeness*, 1979.

[44] S. Gaspers, D. Kratsch, and M. Liedloff. Exponential time algorithms for the minimum dominating set problem on some graph classes. *Algorithm Theory–SWAT 2006*, pages 148–159, 2006.

[45] T. Gerlach and J. Harant. A note on domination in bipartite graphs. *Discussions Mathematicae Graph Theory*, 22(2):229–232, 2002.

[46] F. Glover. *Handbook of Metaheuristics*. Kluwer Academic Publishers, 2002.

[47] F. Grandoni. A note on the complexity of minimum dominating set. *Journal of Discrete Algorithms*, 4(2):209 – 214, 2006.

[48] S. Guha and S. Khuller. Approximation algorithms for connected dominating sets. *Algorithmica*, 20(4):374–387, 1998.

[49] J. Harant, A. Pruchnewski, and M. Voigt. On dominating sets and independent sets of graphs. *Combinatorics, Probability and Computing*, 8(06):547–553, 1999.

[50] F. Harary and T.W. Haynes. Double domination in graphs. *Ars Combin. To Appear.*, 55:201–213, 2000.

[51] T.W. Haynes, S.T. Hedetniemi, and P.J. Slater. *Domination in graphs: advanced topics*. M. Dekker, 1998.

[52] T.W. Haynes, S.T. Hedetniemi, and P.J. Slater. *Fundamentals of domination in graphs*, volume 208. CRC, 1998.

[53] A. Hedar and R. Ismail. Hybrid genetic algorithm for minimum dominating set problem. In David Taniar, Osvaldo Gervasi, Beniamino Murgante, Eric Pardede, and Bernady Apduhan, editors, *Computational Science and Its Applications ICCSA 2010*, volume 6019 of *Lecture Notes in Computer Science*, pages 457–467. Springer Berlin / Heidelberg, 2010.

[54] A. Hedar and R. Ismail. Simulated annealing with stochastic local search for minimum dominating set problem. *International Journal of Machine Learning and Cybernetics*, pages 1–13, 2011.

[55] M.R. Henzinger, S. Rao, and H.N. Gabow. Computing vertex connectivity: New bounds from old techniques. *Journal of Algorithms*, 34(2):222–250, 2000.

[56] C.K. Ho, Y.P. Singh, and H.T. Ewe. An enhanced ant colony optimization metaheuristic for the minimum dominating set problem. *Applied Artificial Intelligence*, 20(10):881–903, 2006.

[57] X. Hou and D. Tipper. Gossip-based sleep protocol (gsp) for energy efficient routing in wireless ad hoc networks. In *Wireless Communications and Networking Conference, 2004. WCNC. 2004 IEEE*, volume 3, pages 1305–1310. IEEE, 2004.

[58] H.B. Hunt, M.V. Marathe, V. Radhakrishnan, S.S. Ravi, D.J. Rosenkrantz, and R.E. Stearns. Nc-approximation schemes for np-and pspace-hard problems for geometric graphs. *Journal of Algorithms*, 26(2):238–274, 1998.

[59] Gurobi Optimization Inc. Gurobi optimizer reference manual, 2013.

[60] R. Niedermeier J. Alber, N. Betzler. Experiments on data reduction for optimal domination in networks. *Annals of Operations Research*, 146:105–117, 2006.

[61] D.S. Johnson, C.R. Aragon, L.A. McGeoch, and C. Schevon. Optimization by simulated annealing: An experimental evaluation; part i, graph partitioning. *Operations Research*, 37(6):pp. 865–892, 1989.

[62] D.S. Johnson, C.R. Aragon, L.A. McGeoch, and C. Schevon. Optimization by simulated annealing: an experimental evaluation; part ii, graph coloring and number partitioning. *Operations research*, pages 378–406, 1991.

[63] R. Jovanovic, M. Tuba, and D. Simian. Ant colony optimization applied to minimum weight dominating set problem. In *Proceedings of the 12th WSEAS international conference on Automatic control, modelling & simulation*, ACMOS'10, pages 322–326, Stevens Point, Wisconsin, USA, 2010. World Scientific and Engineering Academy and Society (WSEAS).

[64] A. Kanevsky. On the number of minimum size separating vertex sets in a graph and how to find all of them. In *Proceedings of the first annual ACM-*

SIAM symposium on Discrete algorithms, pages 411–421. Society for Industrial and Applied Mathematics, 1990.

[65] J.M. Keil. The complexity of domination problems in circle graphs. *Discrete Applied Mathematics*, 42(1):51–63, 1993.

[66] L. L. Kelleher. *Domination in Graphs and its Application to Social Network Theory.* PhD thesis, Northeastern University., 1985.

[67] L.L. Kelleher and M.B. Cozzens. Dominating sets in social network graphs. *Mathematical Social Sciences*, 16(3):267–279, 1988.

[68] D. Kempe, J. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 137–146. ACM, 2003.

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

[70] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

[71] T. Kloks and D. Kratsch. Listing all minimal separators of a graph. *SIAM Journal on Computing*, 27(3):605–613, 1998.

[72] D. Kratsch. Finding dominating cliques efficiently, in strongly chordal graphs and undirected path graphs. *Discrete Mathematics*, 86(1-3):225–238, 1990.

[73] D. Kratsch, P. Damaschke, and A. Lubiw. Dominating cliques in chordal graphs. *Discrete Mathematics*, 128(1):269–275, 1994.

[74] D. Kratsch and M. Liedloff. An exact algorithm for the minimum dominating clique problem. *Theoretical Computer Science*, 385(1-3):226–240, 2007.

[75] Y. Li, Y. Wu, C. Ai, and R. Beyah. On the construction of k-connected m-dominating sets in wireless networks. *Journal of combinatorial optimization*, 23(1):118–139, 2012.

[76] H.I. Lu and R. Ravi. Approximating maximum leaf spanning trees in almost linear time. *Journal of Algorithms*, 29(1):132–141, 1998.

[77] R.D. Luce. Connectivity and generalized cliques in sociometric group structure. *Psychometrika*, 15(2):169–190, 1950.

[78] A. Lucena, N. Maculan, and L. Simonetti. Reformulations and solution algorithms for the maximum leaf spanning tree problem. *Computational Management Science*, 7(3):289–311, 2010.

[79] K. Menger. Zur allgemeinen kurventheorie. *Fundamenta Mathematicae*, 10(1):96–115, 1927.

[80] K. Mohammed, L. Gewali, and V. Muthukumar. Generating quality dominating sets for sensor network. In *Computational Intelligence and Multimedia Applications, 2005. Sixth International Conference on*, pages 204–211. IEEE, 2005.

[81] R.J. Mokken. Cliques, clubs and clans. *Quality & Quantity*, 13(2):161–173, 1979.

[82] M. Morgan and V. Grout. Finding optimal solutions to backbone minimisation problems using mixed integer programming. In *Proceedings of the 7th International Network Conference (INC 2008)*, pages 53–64, 2008.

[83] J. Pattillo, A. Veremyev, S. Butenko, and V. Boginski. On the maximum quasi-clique problem. *Discrete Applied Mathematics*, 2012.

[84] J. Pattillo, N. Youssef, and S. Butenko. On clique relaxation models in network analysis. *European Journal of Operational Research*, 2012.

[85] J. Plesnik and A. Wawruch. On the complexity of finding degree constrained subgraphs. *Acta Mathematica Universitatis Comenianae*, 40:215–218, 1982.

[86] A. Qayyum, L. Viennot, and A. Laouiti. Multipoint relaying for flooding broadcast messages in mobile wireless networks. In *System Sciences, 2002. HICSS. Proceedings of the 35th Annual Hawaii International Conference on*, pages 3866–3875. IEEE, 2002.

[87] V. Raman, S. Saurabh, and S. Srihari. Parameterized algorithms for generalized domination. *Combinatorial Optimization and Applications*, pages 116–126, 2008.

[88] D. Rautenbach and L. Volkmann. New bounds on the k-domination number and the k-tuple domination number. *Applied mathematics letters*, 20(1):98–102, 2007.

[89] C. Ribeiro. *Essays and Surveys in Metaheuristics*. Kluwer Academic Publishers, Norwell, MA, USA, 2002.

[90] L. A. Sanchis. Experimental analysis of heuristic algorithms for the dominating set problem. *Algorithmica*, 33:3–18, 2002.

[91] L.A. Sanchis. Test case construction for the vertex cover problem. *Computational Support for Discrete Mathematics*, pages 315–326, 1994.

[92] L.A. Sanchis. Generating hard and diverse test sets for np-hard graph problems. *Discrete Appl. Math.*, 58(1):35–66, 1995.

[93] S.B. Seidman and B.L. Foster. A graph-theoretic generalization of the clique concept. *Journal of Mathematical sociology*, 6(1):139–154, 1978.

[94] L. Simonetti, A. Salles da Cunha, and A. Lucena. The minimum connected dominating set problem: formulation, valid inequalities and a branch-and-cut algorithm. *Network Optimization*, pages 162–169, 2011.

[95] I. Stojmenovic, M. Seddigh, and J. Zunic. Dominating sets and neighbor elimination-based broadcasting algorithms in wireless networks. *Parallel and Distributed Systems, IEEE Transactions on*, 13(1):14–25, 2002.

[96] M.T. Thai, F. Wang, D. Liu, S. Zhu, and D.Z. Du. Connected dominating sets in wireless networks with different transmission ranges. *Mobile Computing, IEEE Transactions on*, 6(7):721–730, 2007.

[97] M.T. Thai, N. Zhang, R. Tiwari, and X. Xu. On approximation algorithms of k-connected m-dominating sets in disk graphs. *Theoretical Computer Science*, 385(1-3):49–59, 2007.

[98] A. Veremyev and V. Boginski. Identifying large robust network clusters via new compact formulations of maximum k-club problems. *European Journal of Operational Research*, 2011.

[99] F. Wang, H. Du, E. Camacho, K. Xu, W. Lee, Y. Shi, and S. Shan. On positive influence dominating sets in social networks. *Theoretical Computer Science*, 412(3):265–269, 2011.

[100] F. Wang, M.T. Thai, and D.Z. Du. On the construction of 2-connected virtual backbone in wireless networks. *Wireless Communications, IEEE Transactions on*, 8(3):1230–1237, 2009.

[101] D.B. West. *Introduction to graph theory*, volume 2. Prentice hall Englewood Cliffs, 2001.

[102] J. Wu and F. Dai. A generic distributed broadcast scheme in ad hoc wireless networks. *Computers, IEEE Transactions on*, 53(10):1343–1354, 2004.

[103] J. Wu, M. Gao, and I. Stojmenovic. On calculating power-aware connected dominating sets for efficient routing in ad hoc wireless networks. In *Parallel Processing, International Conference on, 2001.*, pages 346–354. IEEE, 2001.

[104] Y. Wu and Y. Li. Construction algorithms for k-connected m-dominating sets in wireless sensor networks. In *Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing*, pages 83–90. ACM, 2008.

[105] D. Yuan. Energy-efficient broadcasting in wireless ad hoc networks: performance benchmarking and distributed algorithms based on network connectivity characterization. In *Proceedings of the 8th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems*, pages 28–35. ACM, 2005.

[106] Z. Yuanyuan, X. Jia, and H. Yanxiang. Energy efficient distributed connected dominating sets construction in wireless sensor networks. In *Proceedings of the 2006 international conference on Wireless communications and mobile computing*, pages 797–802. ACM, 2006.

[107] Z. Zhang, X. Gao, W. Wu, and D.Z. Du. Ptas for minimum connected dominating set in unit ball graph. *Wireless Algorithms, Systems, and Applications*, pages 154–161, 2008.

APPENDIX

Dominating s-club and k-connected d-dominating set considered in Chapter 3 and 4 respectively are the extensions of CDS. The integer programming formulations for minimum dominating s-club and minimum k-connected m-dominating set problem can be used to solve the minimum CDS problem. Here, we briefly review two known mathematical programming formulations for minimum connected dominating set problems. The integer programming formulation for minimum k-connected m-dominating set problem shown in section 4 is a direct extension of one the two formulations for MCDS problem.

It is only recently that integer programming formulation of MCDS appeared in literature. Simonetti et al. [94] present an integer programming formulation and computational experiment results on some standard benchmark test instances. Given a set $S \subset V$, let $E(S) = \{(i,j) \in E | i,j \in S\}$ be the subset of edges of E with both endpoints in S. [94] uses the following decision variable: $y_i \in \{0,1\}, i \in V$: to select which vertices are to be included ($y_i = 1$) or not ($y_i = 0$) in CDS; $x_e \in \{0,1\}, e \in E$: to choose edges that guarantee that the dominating set is indeed connected. Assume that $\mathbb{B} = \{0,1\}$ and $\mathbb{R}$ denotes the set of real numbers. Then the IP formulation from [94] is:

$$\min\{\sum_{i \in V} y_i : (x,y) \in P(G) \cap (\mathbb{R}^m_+, \mathbb{B}^n)\}, \tag{6.9}$$

where polyhedral region $P(G)$ is implied by:

$$\sum_{e \in E} x_e = \sum_{i \in V} y_i - 1 \tag{6.10}$$

$$\sum_{e \in E(S)} x_e \leq \sum_{i \in S \setminus \{j\}} y_i, S \subset V, j \in S \tag{6.11}$$

$$\sum_{j \in N[i]} y_j \geq 1, i \in V \tag{6.12}$$

$$x_e \geq 0, e \in E \tag{6.13}$$

$$0 \leq y_i \leq 1, i \in V. \tag{6.14}$$

The idea behind the formulation above is to use variables x to select edges that guarantee that a spanning tree must be found in the subgraph $G[S]$ for a dominating set $S \subset V$. Constraint (6.10) guarantees that the number of selected edges is exactly one unit less than the number of vertices in CDS. Generalized subtour breaking constraint (GSEC) (6.11) guarantees that the selected edges imply a tree. Constraint (6.12) guarantees that the set of vertices selected form a dominating set.

Vertex-cut based formulation for MCDS. Yuan [105] present another integer programming formulation for minimum CDS problem.

$$\gamma_c(G) = \min \sum_{v \in V} x_v \tag{6.15}$$

$$\sum_{v \in C} x_v \geq 1 \text{ for every minimal vertex-cut } C \subset V \tag{6.16}$$

$$x_v \in \{0, 1\} \ \forall v \in V \tag{6.17}$$

The formulation is still valid when we replace the constraints 6.16 for every vertex-cut. However, the minimal vertex-cuts subsumes the non minimal vertex-cut. Notice that the above formulation could have exponentially many constraints since there exist k-connected graphs with $\Omega(2^k \frac{n^2}{k^2})$ minimum vertex-cuts [64].