NETWORK-CENTRIC METHODS FOR HETEROGENEOUS MULTIAGENT SYSTEMS

A Thesis
Presented to
The Academic Faculty

by

Waseem Abbas

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Electrical and Computer Engineering

Georgia Institute of Technology
December 2013

NETWORK-CENTRIC METHODS FOR HETEROGENEOUS MULTIAGENT SYSTEMS

Approved by:

Professor Magnus Egerstedt, Advisor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Christopher J. Rozell
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Jeff Shamma
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Constantine Dovrolis
School of Computer Science
Georgia Institute of Technology

Professor Anthony Yezzi
School of Electrical and Computer Engineering
Georgia Institute of Technology

Date Approved: 13 November 2013

Dedicated to my parents,

Imdad Hussain and Shahida Parveen.

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my advisor Dr. Magnus Egerstedt for his guidance and direction throughout my research. Without his constant help, this work would not have been possible. He granted me freedom in my research, and encouraged me to explore new avenues. I thank him for helping me learn various aspects of academic research. I also learned from him the skills required to be an effective teacher and an inspiration to students. His invaluable help and guidance made my research a delightful experience.

I am also very grateful to all the members of my reading committee including Dr. Jeff Shamma, Dr. Anthony Yezzi, Dr. Chris Rozell, and Dr. Constantine Dovrolis for reading my thesis and providing me with very useful comments that improved the quality of this work. A special thanks goes to Dr. Robin Thomas for his help with the graph-theoretic aspects of this work. My deep gratitude also goes to Dr. Erik Verriest for providing me an opportunity to work with him. His time and knowledge has always been a great resource for generating new ideas.

Many thanks to my current and former GRITS lab-mates for providing me opportunities to discuss my work and results at a number of occasions. Their constructive feedback and comments have always been very helpful. Special thanks to my close friend, and lab-mate Dr. Hassan Jaleel. I wholeheartedly appreciate your support, advice, and friendship throughout good and bad times. Many thanks goes to my dear friend, Rachel Renée. I cherish your friendship and appreciate the support and affection you have always offered me. I thoroughly enjoyed the company of all my Georgia Tech colleagues and friends. Their companionship made my stay a very pleasant one. Thank you to all of them.

I owe an unpayable debt to my parents. The credit of my every success and achievement goes to their love, blessing, and encouragement. Their trust and continuous support have been my biggest strengths that made me come this far in my education. Many thanks to my brothers, Dr. Nadeem Abbas and Naseem Abbas for their love and affection. They have always brought joy and happiness to my life.

Finally, I would also like to acknowledge the financial support provided for my graduate studies by the Higher Education Commission of Pakistan and Fulbright Scholar Program.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

We present tools for a network topology based characterization of heterogeneity in multiagent systems, thereby providing a framework for the analysis and design of heterogeneous multiagent networks from a network structure view-point. In heterogeneous networks, agents with a diverse set of resources coordinate with each other. Coordination among different agents and the structure of the underlying network topology have significant impacts on the overall behavior and functionality of the system. Using constructs from graph theory, a qualitative as well as a quantitative analysis is performed to examine an inter-relationship between the network topology and the distribution of agents with various capabilities in heterogeneous networks. Our goal is to allow agents maximally exploit heterogeneous resources available within the network through local interactions, thus exploring a promise heterogeneous networks hold to accomplish complicated tasks by leveraging upon the assorted capabilities of agents. For a reliable operations of such systems, the issue of security against intrusions and malicious agents is also addressed. We provide a scheme to secure a network against a sequence of intruder attacks through a set of heterogeneous guards. Moreover, robustness of networked systems against noise corruption and structural changes in the underlying network topology is also examined.

CHAPTER I

BACKGROUND AND INTRODUCTION

1.1 Heterogeneous Multiagent Systems

A network of agents, in which individuals with a diverse set of resources or capabilities interact and coordinate with each other to accomplish various tasks, constitutes a heterogeneous multiagent network. In recent years, heterogeneity has emerged as an important aspect of multiagent and cooperative networks. Agents with assorted capabilities and properties, when integrated together in the framework of a heterogeneous network, become specialized to achieve sub-goals efficiently. For instance, it is shown in [1] that the reliability and lifetime of a sensor network can be increased by introducing nodes in the network that are different in terms of power consumption and communication capabilities. In [2], it is shown that heterogeneity can be exploited to reduce the number of sensors required in a sensor network without compromising on the coverage and the broadcast reach of the network. Heterogeneity among agents is explored to make a decentralized system more stable and efficient in [3]. Several other applications of such heterogeneous systems have been studied in various domains, including multirobot systems (e.g., [4, 5]), and wireless sensor networks (e.g., [6]), to name a few.

Advantages of having multiple capabilities and resources through heterogeneous agents come at a cost of added complexity within a system that can be traced back to one of the following categories: (a) problem formulation and description in terms of heterogeneous components, (b) inter-communication and inter-operation capabilities among different agents, (c) inter-connection topology of a system, and (d) overall efficiency and performance of the system. All of these issues are inter-related, and

influence each other as shown in Fig 1.1.

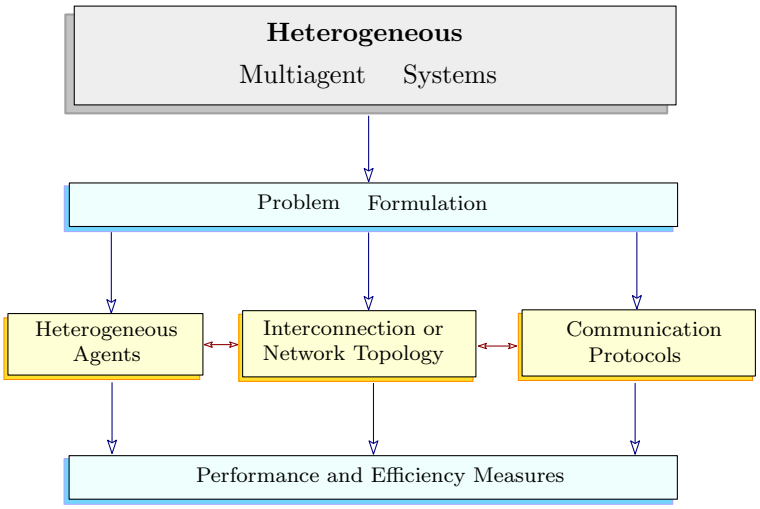


Figure 1.1: Components of a typical heterogeneous multiagent system.

One of the challenges in heterogeneous networks is to optimally distribute agents having different capabilities and resources. A part of the challenge is to articulate the notion of *heterogeneity* in order to guide the analysis and design of heterogeneous networks in a systematic way. Heterogeneity in cooperative networks can be understood along a number of dimensions. We can broadly classify the studies in this area into two categories; one that quantify heterogeneity from *agents' perspectives* including functional or behavioral dissimilarities among individual agents (e.g., [7, 8, 9]), and the relative number of non-homogeneous agents (e.g., [10, 11, 12]); the second category measures heterogeneity in terms of the *underlying graph structure* of the network with an aim to quantify the degree distribution among the nodes while treating all the nodes similar[1] (e.g.,[13, 14]).

The quantitative aspect of heterogeneity in the context of classification of agents deals with the relative number of agents of different types that exist within a system. A number of measures have been proposed to quantify this aspect. In economics,

[1]i.e., there is no distinction among the nodes to account for their functional or behavioral differences

some popular measures to quantify the differences among population include, range ratio, McLoone index, Atkinson index, coefficient of variation, Gini coefficient, and Theil index (e.g., see [10, 11]). Each of these indices has its own merits and demerits based on specific circumstances. These measures are significant towards analyzing the effectiveness of various economic policies. In ecological and biological systems, popular diversity measures include, species richness, functional group richness, evenness, Shannon index, Simpson index, Gini-Simpson index, and Berger-Parker index (e.g., [8, 15, 16]). All of these measures quantify in some way the number of different species represented in a collection of individuals. These metrics are effectively used to examine the effects of biodiversity on ecosystem functioning and services (e.g., [15, 17]). Similarly, heterogeneity in social networks has been exploited through various indices including Herfindahl index and other entropy based diversification measures (e.g., [18]). It is to be mentioned here that in all these metrics, the notion of heterogeneity in a network is quantified in terms of the relative number of various types of agents, species, or populations. The functional differences among individuals in a system are not accounted for.

To measure the qualitative differences among various classes of individuals within a system, several other measures have been proposed and analyzed. These measures quantify the functional or behavioral heterogeneity among agents, i.e., how are individuals different from each other? In ecological research, functional diversity accounts for the range of things various organisms and species can do in communities and ecosystems. Based on the choice of functional traits with which individuals can be distinguished (e.g., variation in the functional characters, the complexity of food webs, interaction with their environment and with each other, utilization of resources, taxonomy etc.), various indices of functional diversity have been developed to study and predict the impact of species on ecosystem and vice versa. These indices can be broadly classified into the three categories namely, functional richness, functional

evenness, and functional divergence (e.g., see [8]). These measures incorporate the effect of complementarity and redundancy of co-occurring species to analyze the reliability of ecosystem processes [29].

Rao's quadratic entropy [7, 30] (along with its variants) is also a popular and widely used measure to quantify the degree of difference among individuals of different species [9, 31]. The idea is to calculate the expected distance among any two randomly selected individuals, where the distance function can be defined in many different ways considering one or more characters of species (e.g., [16]). Other dissimilarity measures include Hendrickson and Ehrlich index [34], which is a non-biased version of Rao's diversity coefficient; Warwick and Clarke index [35]; Gower similarity coefficient [36]; and other taxonomic diversity measures [37, 38], in which innovations are made by introducing differences between species. Several information theoretic measures, based on Shannon entropy, have also been studied to quantify diversity and the degree of difference among species (e.g., see [39, 40]).

From the perspective of the underlying graph structure of the network, degree distribution among the nodes within the network has been the central theme to quantify network heterogeneity. Degree of a node v in a network is the number of nodes that are directly connected or adjacent to v. If n_k is the number of nodes having a degree k, then the probability of randomly selecting a node with a degree k is $P_k = \frac{n_k}{n}$, where n is the total number of nodes. Degree distribution is the probability distribution of these degrees over the whole network, and is a significant characteristic of many network models such as scale free networks (having a degree distribution of power law form), and the small world models (having an exponential degree distribution). Many real world networks do not exhibit a well defined degree distributions such as exponential or power law. Thus, various indices have been proposed to quantify the heterogeneity of such networks in terms of the degree distribution. Some of the examples include degree variance, Collatz and Sinogowitz irregularity index, Albertson

index, and Estrada index (e.g., see [33, 13, 14]. Entropy based heterogeneity measures to quantify the degree inequality in graphs have also been proposed (e.g., see [32]). It is to be mentioned here that all these measures aim to quantify the structure of the network through the distribution of node degrees while treating all the nodes as similar, i.e., there is no distinction among the nodes that accounts for their functional or behavioral differences.

Similarly, the issue of heterogeneity in multirobot systems has been addressed under various contexts. In [4], a distributed and behavior-based architecture is proposed for heterogeneous multirobot networks, where the architecture allowed robots to continually change their capabilities and update their performance for the completion of assigned tasks. In [12], the concept of hierarchic social entropy, which is an application of Shanon's information entropy, is introduced to quantify diversity in a robot team, and at the same time measure behavioral differences among robots. A dynamic role assignment based approach to distributed coordination in heterogeneous multirobot systems is presented in [41]. The approach relies on the broadcast of utility functions, which define the capability of every robot to perform a task. The significance of heterogeneity in the control systems of robots is highlighted in [42], in which a team of robots is allowed to co-evolve their controllers for solving tasks with various difficulty levels. In [43], a computational framework is presented to model heterogeneous agents as well as the environment in which they are situated. Several other studies have also explored heterogeneity in multirobot systems at behavioral (e.g., [44, 45]) as well as hardware level (e.g., [46]).

Despite their merits, all these studies address one of the two attributes of cooperative networks at a time, either agents' classification or the topological properties of the underlying graph structure. However, to obtain a holistic view of heterogeneity in networks in which agents with a diverse set of capabilities coordinate with each other,

a unified framework that incorporates agents' classification as well as network topology is needed. Similarly, in the context of resource allocation, coordination among agents of various types and the structure of the underlying network has a profound impact on the overall behavior and functionality of the system. In this work, we focus on the issue of exploiting a connection between the structure of the information exchange network and the diverse set of resources contained by agents within a network. We approach this issue by developing an inter-connection topology framework for heterogeneous multiagent systems with an aim to explore heterogeneity of its elements in a constructive way [20, 21, 27]. Various aspects of heterogeneity in multiagent networks are summarized in Fig. 1.2

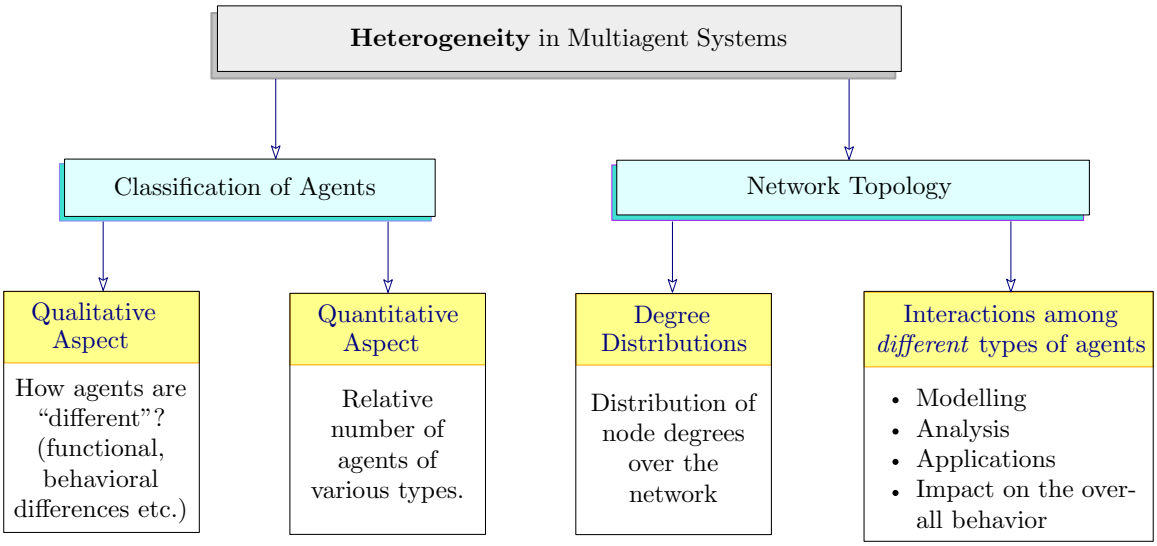


Figure 1.2: Various aspects of heterogeneity in multiagent systems.

1.2 Domination in Graphs

Several concepts from domination theory in graphs are employed in this work to formalize and study network topology based heterogeneity in cooperative networks. Domination in graphs is an extensively studied domain in graph theory. As the concepts of domination, covering, and centrality in graphs are largely inter-related, a number of variants of the concept of domination in graphs have been developed. The

basic object in a graph domination is a *dominating set*.

Definition 1.2.1 *In a graph $G(V, E)$ with a vertex set V and an edge set E, a dominating set is a subset S of vertices such that every vertex in G is either in S or adjacent to S. The minimum number of vertices in any dominating set of a graph is the domination number of the graph.*

As an example, consider a graph $G(V, E)$, in which each vertex represents a city and an edge between two vertices indicates transmission between cities because of a transmission tower that is located on one of the cities. The goal is to find the minimum number of transmitting towers so that every city either has a transmission tower, or is in the range of at least one transmission tower. The minimum number of required towers is equal to the domination number of the corresponding graph, and the towers need to be located on the vertices in the dominating set. Fundamental concepts related to the notion of domination along with various applications are detailed in [47].

The notion of domination in graphs finds a number of applications in many different fields, including design of efficient algorithms for routing in ad hoc wireless networks (e.g., [50]), coverage optimization in sensor networks (e.g., [51]), energy conservation in sensor networks (e.g., [52, 53]), facility location problems in operations research (e.g., [54]), and information propagation in social networks (e.g., [57]), to name a few. As a result of the wide applicability of domination related concepts in various fields, multitude of variants of domination have been introduced by adding various constraints and restrictions on the dominating set. Examples of domination in graphs include, *conected domination*, in which all the vertices in a dominating set must induce a connected sub-graph; *k-distance domination*, in which every vertex in a graph must lie within a distance $k \geq 0$ of some vertex within a dominating set; *multiple domination*, in which every vertex in a graph must be dominated by at least

k vertices in a dominating set for some $k \geq 1$; and many more (e.g., [47, 48, 49]). The basic idea among all these variants remains the same, i.e., to figure out a subset of vertices within a graph that *dominates* the whole vertex set in a specified way. We can broadly classify research in the field of domination in graphs into one of the categories shown in Fig. 1.3.

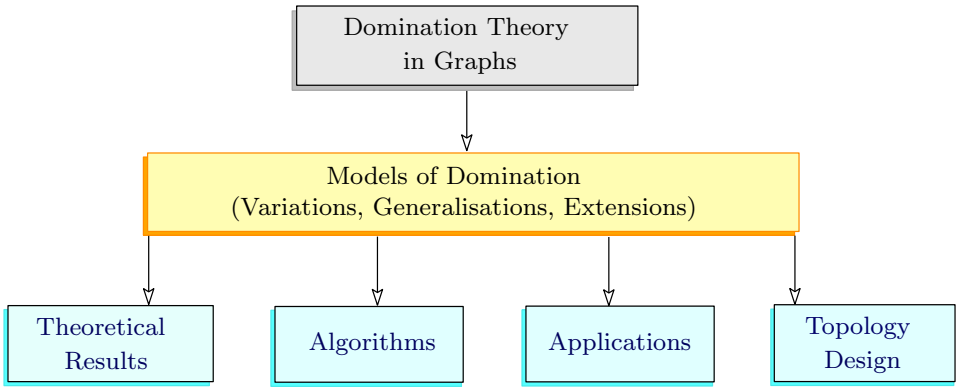


Figure 1.3: Areas of research in the field of domination in graphs.

Results regarding theoretical aspects include, bounds on the domination number, extending and generalizing various notions of domination, developing new mathematical techniques, and analysis of various domination related parameters. The problem of determining the domination number of a graph is known to be NP-complete for an arbitrary graph [58], which suggests a need for a thorough study and discussion of the complexity issues and algorithmic aspects of domination in graphs. All these studies make it possible to employ various domination related concepts for the analysis and design of network topologies that are optimal in some sense for a wide range of applications.

Another related notion of significant interest is the maximum number of disjoint dominating sets in a graph, referred to as the *domatic number* [59].

Definition 1.2.2 *A domatic partition of a graph $G(V, E)$ is a partition of V into disjoint sets, $V_1, V_2, \cdots, V_k$, such that each V_i is a dominating set. The domatic number is the maximum size of a domatic partition.*

The concept of domatic partition is particularly useful in situations, in which multiple facilities should be assigned to nodes in a manner that allows every node to find all different types of facilities in its closed neighborhood. For example, let there be a group of villages connected with each other through a network of roads. Every village can be accommodated with a single facility from a set of facilities, say $F = \{f_1, f_2, \cdots, f_k\}$. The objective is to distribute these facilities such that every villager can find all of these facilities without travelling any farther from its neighbor village. The maximum number of facilities that can be accommodated within this network is equal to the domatic number of the underlying graph. Therefore, a domatic number determines the *capacity* of a network to accomodate different types of resources in such problems. The domatic partition problem is one of the classic NP-hard problems [60]. The domatic number for various graph families have been studied, including regular graphs [61], interval graphs [62], circular arc graphs [63], strongly chordal graphs [64], and others (e.g., see [65]).

As the types of resources, which are assigned to various nodes within a network, exceed the domatic number of the network, there will always be a node missing at least one resource type in its closed neighborhood. To deal with such situations, multiple types of resources are assigned to a node. For instance, s different types of resources can be assigned to a vertex instead of only one. Under such an arrangement, ks types of resources can be assigned to a network with a domatic number of k. However, it is shown in [66] that it may be possible to incorporate *more* than ks types of resources in a network whose underlying graph has a domatic number of k. Thus, the ability of a network to accommodate heterogeneous entities may improve significantly with the leverage of assigning multiple resources to nodes.

In this work, we make use of these domination related concepts to model network topology based heterogeneity in multiagent systems. In addition to the existence of various types of agents, interactions and coordination among agents will be crucial in

characterizing and exploiting heterogeneity in our model.

1.3 Exploiting Heterogeneity in Multiagent Systems

In a heterogeneous multi agent system, an agent locally performs some task by interacting with its neighbors. In terms of the network topology based heterogeneity, the extent and efficacy of the task depends on the number of heterogeneous components involved in the task, that is to say how many different types of agents are present in the neighborhood of an agent. In a network with r different types of agents, if an agent v can interact with all r types in its closed neighborhood, heterogeneity of the task performed by v will be maximal because v can exploit all r different functionalities available in its closed neighborhood. For the same reason, heterogeneity of the task by an agent u, which has only one type of node available in its closed neighborhood, will be minimum. If all of the nodes in a network are capable of performing a maximally heterogeneous task, i.e., every node in a network has all r types of nodes in its closed neighborhood, then the network will be completely heterogeneous with r types of resources or capabilities [21].

1.3.1 Energy Efficient Heterogeneous WSANs

Wireless sensor and actor networks (WSANs) provide an effective solution to the distributed sensing and response related problems. In such networks, information gathered by the sensing nodes is made available to the actor nodes through a wireless medium that utilize this information to make decisions and act upon the environment. A wide variety of applications of WSANs have been reported in environmental monitoring, surveillance frameworks, attack detection, and manufacturing automation systems to name a few (e.g., [69, 70]).

Heterogeneity emerges as an important property of WSANs, in which sensors with a varying set of sensing and transmission capabilities are deployed within some field of interest. For instance, for the purpose of environment modeling, a set of temperature,

air flow, and pressure sensors may be deployed in the field to observe various aspects of the climate. Moreover, an actor present in the field may not have access to all sorts of measurements by the sensors. Thus, actors rely on coordination and communication with each other to acquire complete information under the automated architecture of WSANs [69]. At the same time, managing the activity of sensor nodes through efficient activity scheduling mechanisms is imperative for a longer lifetime of the sensor network as they have limited power resources.

In this work, we address the issue of energy-efficient information gathering in heterogeneous WSANs. A scheme is proposed in which actors utilize coordination among themselves, and heterogeneity among sensors to deactivate a large portion of originally deployed sensors of various types to preserve sensors' power resources [25]. Under the initial deployment of sensors, if an actor v receives a particular type of data either directly from a sensor, or by interacting with a neighbor actor, then as a result of our scheme, v continues to receive same data but with only a small subset of activated sensors. We formulate this problem in graph-theoretic terms, thereby providing solutions using graph-coloring and graph-domination related concepts.

1.3.2 Efficient Complete Coverage Through Heterogeneous Disks

Coverage problems constitute an important class of problems in the domain of multiagent and multirobot systems. The primary objective is to deploy and distribute agents with sensing capabilities to completely monitor a domain under consideration while satisfying certain constraints and criteria. Coverage problem can find its roots in computational geometry (e.g., [67, 68]). One of the primary problems there is related to circle covering[2]; what is the most *efficient* way to place circular disks of same radii to completely cover a region? The solution is provided in [67] stating that the most efficient way that minimizes the overlap among disks is to place disks

[2]A circle covering is a configuration of overlapping circles with given radii to completely cover some domain

in an equilateral triangle lattice. In multiagent and sensor networks, circular disks correspond to the sensing footprints of sensing nodes and minimizing overlap means that power is consumed efficiently for the sensing operation.

In this work, we examine if efficiency of the circle coverings can be further increased by using heterogeneous circular disks, i.e., disks with different radii. In fact, it is shown that for complete coverage, configurations of heterogeneous disks exhibit better efficiency as compared to the homogeneous case [28].

1.4 Eternal Security in Graphs

Security and protection against malicious agents and external intrusions is often required for a reliable operation of a network. Anomalous behavior of an individual agent may result into an abnormal or inconsistent behavior of the overall system. This situation demands a proper surveillance of the system as well as an efficient response mechanism to counter any anomalies within the network. Problems related to search and secure scenarios have been studied in the literature for various domains, including computer networks, information systems, multi agent and multirobot systems, to name a few. Under various settings, the objective remains the same, i.e., to *detect* an external intrusion or an internal abnormality among agents, and effectively *respond* to these anomalies. A number of schemes and strategies have been reported to address these issues for various distributed and cooperative networks. These schemes include cooperative minimum time surveillance algorithms [71], communication and connectivity constrained surveillance problems (e.g., [72, 73]), distributed detection schemes for observing abnormalities within agents (e.g., [74, 75]), the number of guards required for monitoring all agents (e.g., [76]), cooperative path planning for security applications [77], and pursuit-evasion problems for security applications under various constraints (e.g., [78, 79, 80]).

Since a multiagent network is often modeled by a graph, the idea of network

protection against malicious agents can be associated to the notion of security in graphs in some way. A graph-theoretic interpretation of search and secure problems, initially addressed in [81], has been extensively studied in the literature (e.g., see [82, 83]). In this proposal, we employ tools from domination theory in graphs along with the notion of *heterogenous guards* to secure a network of agents against an infinite sequence of intruder attacks. The concept of *eternal security in graphs*, introduced in [84], is a variant of domination in graphs. It addresses the problem of making all the nodes in a graph secure against an infinite sequence of intruder attacks by a certain minimum number of guards, which are located on the vertices of a graph. An *intruder attack* on a node (or a vertex) refers to any malicious activity on that node. A *guard* is an agent that can detect and respond to an intruder attack within its range by moving from one node to another along the edges of a graph. If these guards are located in a graph in such a way that every node lies within a range of at least one guard, the graph is secured against an intruder attack on any of its nodes. We may call this location of guards as a *secure configuration*. The movement of a guard from one node to another to neutralize a threat may disturb this secure configuration, and leave a subset of nodes un-secured since these nodes may not lie in the range of any guard in the updated configuration.

As an example of eternal security, consider a graph shown in Fig 1.4(a), in which two guards g_1 and g_2 are securing the vertices of the graph where each guard can detect and respond to an attack on an adjacent vertex only. In the case of an attack on a vertex indicated by an arrow, guard g_1 moves towards it, which results in an un-secure configuration since the circled vertices have no guard in their neighborhoods. In this example, the problem is that the *number of guards* is not sufficient. In Fig 1.4(b), there are three guards g_1, g_2, and g_3. Each guard is capable of detecting and responding to an attack on an adjacent vertex. However, after two intruder attacks, the configuration of the guards is unable to secure all the vertices. Though the

number of guards is sufficient in this case, the *strategy* for the movement of guards to counter intruder attacks is not appropriate. In Fig 1.4(c), the number of guards is still three but the movement strategy of guards to counter intruder attacks is such that the resulting configuration is always secure, i.e., for every vertex there *always* exists a guard to secure it. This makes the graph eternally secure against a sequence of intruder attacks.

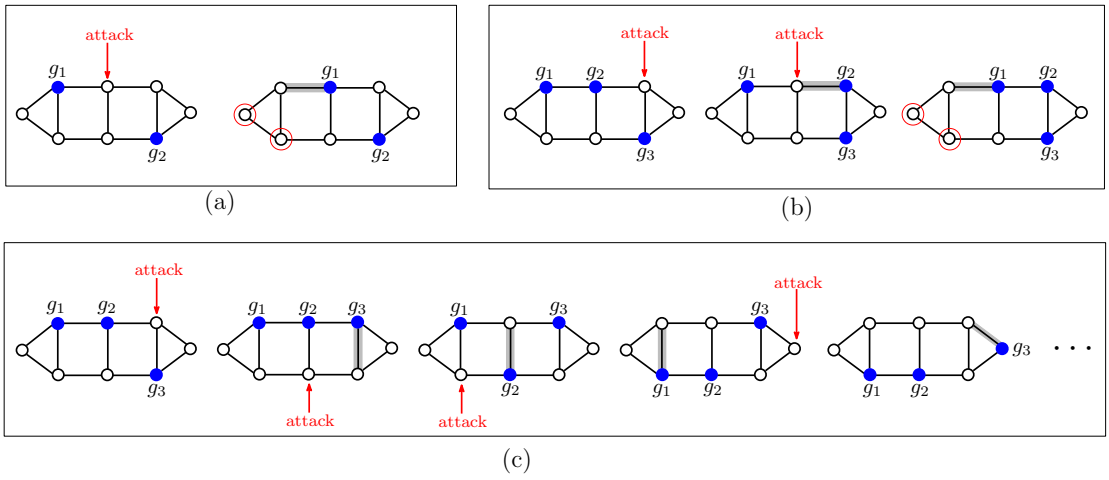


Figure 1.4: (a) Two guards are not sufficient to *eternally secure* a graph. (b) The movement of guards result into an unsecured configuration after two intruder attacks. (c) The graph is eternally secured by three guards.

The idea behind eternal security is to secure vertices against an arbitrary sequence of attacks. The objective is to determine the number of guards for given ranges, deploy them within a graph, and outline a strategy for guards such that they always maintain a secure configuration even after their displacement among various nodes. These issues are highlighted in Fig. 1.5. Various results regarding the number of guards required for the eternal security of graphs have been reported in the literature. In [56], number of guards required for the eternal security is related to the domination number in graph, whereas in [85, 86], bounds on the number of guards are provided in terms of the independence number of a graph. The relationship between the vertex cover number and the number of guards is explored in [87]. In all these

results, a guard can detect and respond to an intruder attack at a vertex that lies in the immediate neighborhood of the guard, i.e., adjacent to the guard. At the same time, all these studies focus on the number of guards required for the eternal security without providing strategies for the movement of guards in the case of intruder attacks. However, sole knowledge of sufficient number of guards is not enough for the eternal security. The location of guards and their right moves is also crucial for securing graph nodes against a sequence of intruder attacks. Moreover, a generic case of guards with non-homogeneous ranges also needs to be investigated.

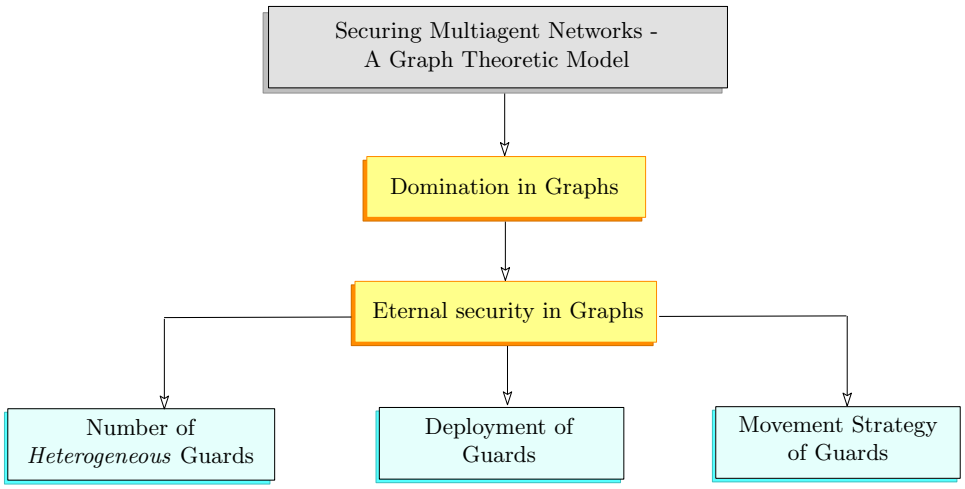


Figure 1.5: The problem of securing a network against a sequence of intruder attacks can be solved using the notion of eternal security in graphs.

Heterogeneous guards are particularly useful when security requirements vary for various nodes. As an example, consider a network that needs to be secured against a possible sequence of external attacks on its nodes. Furthermore, assume that there are certain nodes that are more sensitive or critical than the others, and require an immediate consideration in case of an attack. Thus, a secure configuration of guards is needed at all times with an added constraint that the distance between the critical nodes and the guards should be lesser than a certain value, say r_1 path lengths, whereas the distance between the guards and the remaining (non-critical) nodes can be relatively large, say $r_2 \geq r_1$. In this scenario, eternal security of a graph

through a set of guards having different ranges (r_1 and r_2) can be more effective and useful as compared to the security of a graph through a set of homogeneous guards. Furthermore, the energy consumption of guards varies with their ranges (e.g. a guard with a longer range consumes more energy), so an energy-efficient security solution is also possible when guards with different ranges are deployed within the system.

In this work, we study the issue of eternal security in graphs in a systematic way by connecting together all of its major components, including number of guards, deployment of guards, and their movement strategy in case of an attack [22]. An algorithm, specifying an appropriate placement as well as movement strategies for guards in case of intruder attacks, is also presented. Eternal security in graphs through heterogeneous guards, which may have different ranges from each other, is also investigated.

1.5 Robustness Issue in Networked Systems

The issue of making networks reliable and resilient is one of the integral matters of the network design process. In Section 1.4, we discussed one aspect of this broader issue of reliability in networks in terms of securing a multiagent network against intruder attacks and malicious agents. Another dimension of reliability and consistent performance is the robustness of networked systems against noise corruption and to the structural changes in an underlying network topology. Robustness in networked systems can be further studied from two different perspectives. Firstly, how well a system behaves in the presence of noise, i.e. robustness against noise or *functional robustness*, and secondly what is the effect of change in network topology (due to edge or node failures) on the performance of such systems, i.e., *structural robustness*. Both of these aspects have been studied in the literature and various indices have been proposed to measure them. Edge (vertex) connectivity, algebraic connectivity as introduced in [88], betweenness [89], information centrality, toughness and other spectral measures (e.g., see [97]) are some of the parameters that have been used to

quantify structural robustness in graph structures. Robustness of networks, in which agents implement consensus protocols in the presence of noise, has been addressed by providing various distributed algorithms and schemes to minimize corruption of noise in such systems [95, 96, 93, 91]. Most of the studies on structural robustness and robustness against noise seem to be independent of each other, focusing on either one of the aspects. Here, we show that both of these robustness view-points are in fact, related to each other and therefore, can be measured simultaneously by a same parameter.

In [94], *Kirchhoff index* of a graph is introduced through the notion of effective graph resistance. The idea is to obtain an electrical network from a graph by replacing each edge with a unit resistance. The total electrical resistance between any two nodes in such a network is the effective resistance between the corresponding vertices of a graph. The sum of all pair-wise effective resistances between vertices is the Kirchhoff index, K_f, or the effective resistance of a graph. A network of agents can be modeled by an undirected graph, in which vertices represent agents and edges are the information exchange links among agents. Recently, it is shown in [91] that functional robustness of systems, in which agents update their states by a linear consensus protocol in the presence of additive white noise, can be measured by the Kirchhoff index of the underlying graph of the network. On the other hand, in [90] it is demonstrated that the effect of edge failures on the overall connectivity of a graph can be quantified by the effective graph resistance, which is equivalent to the Kirchhoff index of the graph (as shown in [94]). Thus, both aspects of robustness can be specified by an exactly same graph invariant.

In this work, we further explore this relationship between structural robustness and functional robustness (robustness due to noise) in multiagent systems through the Kirchhoff index of the underlying network topology. We investigate the role of various network topologies on the robustness property of these systems. A systematic

way to construct robust network structures is then provided. In particular, we define a graph process to obtain maximally robust graph structures at each step of the process by an optimal addition edges. Moreover, a relationship between the symmetry of a network structure and its robustness is also discussed [23].

1.6 Organization

All of the issues and problems introduced in this chapter are investigated in detail in upcoming chapters. Heterogeneity related matters in multiagent systems are addressed in Chapters 2, 3, and 4, whereas security and robustness for reliable functioning of networks is the main focus of Chapters 5 and 6. In Chapter 2, a network topology based design of heterogeneous networks is presented using graph-theoretic concepts. The related results are published in [20, 21, 26, 27]. The framework developed is then utilized in Chapter 3 to design an energy efficient data collection scheme in heterogeneous wireless and sensor networks [25]. The value of heterogeneity among agents is explored in Chapter 4 to obtain efficient solutions to the complete coverage problem in multiagent and multirobot systems. The results are documented in [28]. Security of multiagent networks from a sequence of external intrusions is addressed in Chapter 5, and the related results are published in [22]. The notions of structural and functional robustness, and inter-relationship among them are elaborated in Chapter 6, and results are published in [21].

CHAPTER II

NETWORK TOPOLOGY FOR THE DESIGN OF HETEROGENEOUS NETWORKS

Heterogeneity has emerged as an important aspect of multiagent systems, in which agents with different capabilities and resources interact with each other to perform various complex tasks. In this chapter, we aim to provide a framework for a network topology based characterization of heterogeneity in multiagent and cooperative networks that also incorporates distinctions among agents of various types. Coordination and interactions among agents are modeled using constructs from graph theory. A graph coloring problem is then formulated to examine an inter-relationship between the network topology and distribution of agents with assorted capabilities in heterogeneous networks. As a result, we will be able to answer how agents with different resources can be distributed within a network to maximally exploit the available resources through local interactions.

2.1 Graph Coloring Based Model of Heterogeneous Networks

Graph-theoretic tools are frequently applied to model and analyze various cooperative networks including sensor networks, multi-agent and multi-robot networks. A network is modeled by a graph $G(V, E)$ in which the vertex set V represents agents and the edges in the edge set E correspond to interactions among agents. In the case of heterogeneous networks, in which agents may be different from each other in terms of their resources or capabilities (for instance, sensing, actuation, dynamics, capabilities, resources, hardware, or software etc.), heterogeneity can be modeled by associating a unique *color (or label)* with each resource type available in the network. Next, all

of the vertices in an underlying graph of the network are assigned colors (or labels) in accordance with the resources contained by the corresponding agents. A vertex may have multiple labels if the corresponding agent has multiple types of resources. In such heterogeneous networks, agents interact and utilize each others' resources to accomplish various tasks. The availability of resources of different types in the local neighborhood of an agent determines the agent's overall capability to perform various tasks. The inter-relationship between the network topology and the distribution of agents with various types of resources can be studied through the graph coloring formulation in which vertices are assigned labels in accordance with the resources contained by the agents.

Throughout, a *graph* $G(V, E)$ with a vertex set V and an edge set E, is a simple undirected graph. An edge between nodes v_i and v_j is denoted by $v_i \sim v_j$. The *open neighborhood* of a vertex $v \in V(G)$, denoted by $\mathcal{N}(v)$, is the set of vertices adjacent to v. Its *closed neighborhood*, denoted by $\mathcal{N}[v]$, is $\mathcal{N}(v) \cup \{v\}$. The degree of a vertex v, $deg(v)$, is the cardinality of $\mathcal{N}(v)$. The minimum degree of a graph, $\delta(G)$, is $min\{deg(v) \mid v \in V\}$ and the maximum degree of a graph, $\Delta(G)$, is $max\{deg(v) \mid v \in V\}$. We will use the terms color and label interchangeably.

Let $\mathbf{r} = \{1, 2, \cdots, r\}$ be a set of labels that represents r different types of resources (or capabilities) available within a heterogeneous network. Furthermore, the vertices are assigned labels according to the map,

$$f: \quad V \longrightarrow 2^{\mathbf{r}}$$

$2^{\mathbf{r}}$ is the set of all subsets of $\mathbf{r}$. $f(v)$ is a subset of resources contained by agent v, which interacts and utilizes the resources of its neighbors to perform some task. Thus, *heterogeneity* of an agent v within the network depends on the resources (or capabilities) contained by v and its neighbors and can be defined as

$$\mathcal{H}(v) = \bigcup_{u \in \mathcal{N}[v]} f(u) \tag{1}$$

Moreover, an agent is *maximally* heterogeneous within the network whenever $\mathcal{H}(v) = \mathbf{r}$, as v can exploit all different functionalities and resources available within the network by interacting with its neighbors. Thus, from the network topological view-point, we define a *completely heterogeneous* graph as

Definition 2.1.1 *A graph $G(V, E)$ in which every $v \in V$ is assigned a subset of labels $f(v)$ from the set $\mathbf{r} = \{1, 2, \cdots, r\}$, is completely heterogeneous with r labels if*

$$\mathcal{H}(v) = \mathbf{r}, \quad \forall v \in V$$

Agents, which are the individual components in a heterogeneous network, cooperate and complement each others' expertise and resources, thus allowing the overall system to exhibit significantly greater functionalities. In a *completely heterogeneous* network as defined in Definition 2.1.1, every agent is capable of exploiting a complete set of resources and functionalities available within the network to perform various tasks by working in conjunction with its neighbors.

2.1.1 Examples

Consider an industrial location where some manufacturing process depends on environmental conditions, including temperature (t), light (ℓ), humidity (h), and air pressure (p). A specific environmental condition, say $\omega(t, \ell, h, p)$, which depends on all of the above parameters, is needed to be maintained to get a desired yield. Sensors for each of the above parameters t, l, h and p are mounted at various data collection points, which are inter-connected with each other and exchange data. The environmental condition $\omega(t, \ell, h, p)$ is computed at every such data collection point. The distribution of sensors with assorted sensing capabilities constitutes a heterogeneous network. It is further assumed that owing to some constraints (e.g., hardware, power, economical etc.), only a subset of sensors can be mounted at each data collection point. Since all four parameters are needed for the computation of $\omega(t, \ell, h, p)$, sensors need to be distributed in such a way that all of the four types of sensors are

available in the closed neighborhood of every data collection point. In other words, underlying graph of the network needs to be completely heterogeneous with the set of labels $\{t, l, h, p\}$ as shown in Fig. 2.1.

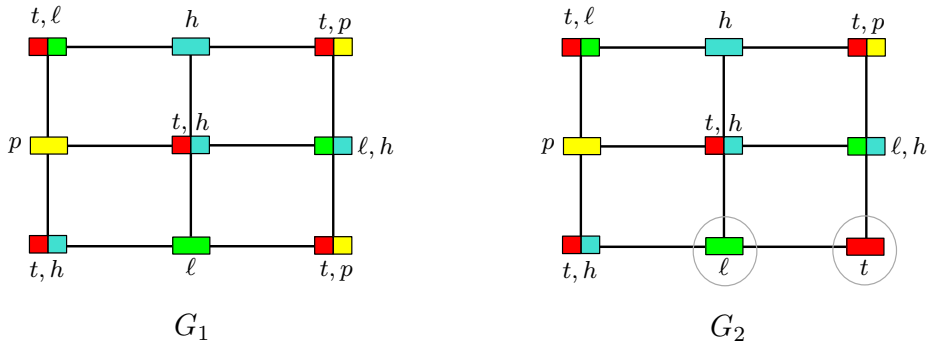


Figure 2.1: G_1 is a completely heterogeneous graph with four labels. The labeling set is $\{t, h, l, p\}$. G_2 is not completely heterogeneous as label p is missing from the closed neighborhoods of circled nodes.

As another example, consider a society of some 'species', in which each member of the society has been assigned a specific role. Some members are food providers, some are shelter providers, while others hold the task of providing security to the members they interact with. In such a society, every member depends on other members to ensure the availability of all the facilities. For instance, a food providing member must interact with a shelter provider and a security provider for shelter and security respectively. This kind of cooperation constitutes a heterogeneous network, in which availability of all the resources to each member of the society is possible if the underlying graph of the network is completely heterogeneous with three distinct labels.

2.1.2 Major Issues Related to the Notion of Completely Heterogeneous Graph

The notion of completely heterogeneous graph has three major aspects in the context of heterogeneous cooperative networks. First, given a colored graph, how can we analyze the distribution of colors to nodes, i.e., determine in a systematic way which

nodes are missing colors in their closed neighborhoods? what is the most deficient color in the network? which edges or nodes are relatively more 'crucial' or significant? This analysis will provide a way to transform a given coloring of a graph to a completely heterogeneous one through simple graph operations, such as adding or removing certain edges.

Second, given a coloring set $\mathbf{r}$, and a constraint on the maximum number of colors each node can have, is it possible to color nodes to get a completely heterogeneous graph? For instance, in Fig. 2.2, it is impossible to color G_1 to get a completely heterogeneous graph with five colors if each node is allowed to have at most two colors. This issue is significant in characterizing network topologies in the sense that it deals with the overall capability of the network to incorporate heterogeneous resources.

Third, once we know that a labeling scheme exists to make a graph completely heterogeneous with a certain number of labels and constraint on the number of labels each node can have, how can we assign colors to nodes to achieve such a labeling scheme? An example is illustrated in Fig. 2.2.

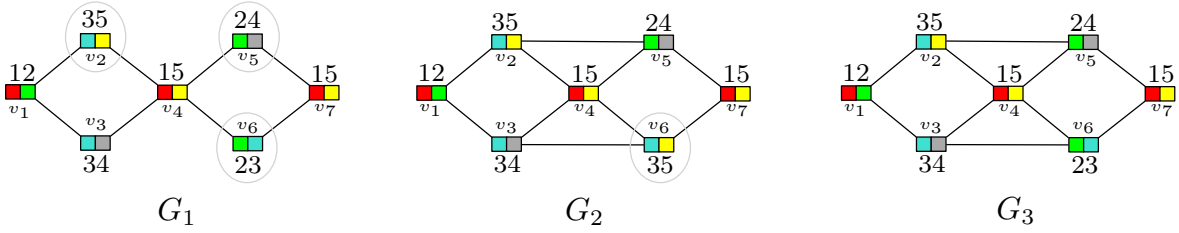


Figure 2.2: G_1 can not be made completely heterogeneous with five labels by assigning a maximum of two distinct labels to each node. Although it is possible to assign two colors to each vertex in G_2 and obtain a completely heterogeneous graph with five colors, v_6 is missing label 2 in its closed neighborhood under the given labeling. In G_3, each vertex has a complete set of five labels in its closed neighborhood.

2.2 Distribution of Heterogeneous Agents within a Network

In this section, we analyze distribution of resources or capabilities in heterogeneous cooperative networks using the model introduced in Section 2.1. Various tasks performed by individual agents in such networks depend on the resources available locally to the agents. Thus, information regarding the missing resources in the closed neighborhoods of agents along with the interactions needed to make these resources available to agents is crucial. We address these issues in this section.

Given a graph with n nodes, in which each node v_i is assigned a subset of labels $f(v_i)$ from the set of labels $\mathbf{r} = \{1, 2, \cdots, r\}$. We define a *color matrix*, denoted by C, as a binary matrix with dimensions $n \times r$ as follows:

$$C_{ij} = \begin{cases} 1 & \text{if } \ j \in f(v_i), \ \text{ where } f(v_i) \subseteq \mathbf{r} \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

In (2), $f(v_i)$ indicates the colors (labels) assigned to the vertex v_i. The column index of C indicates the color (label), thus $C_{ij} = 1$ means that the color j has been assigned to the vertex v_i.

Using the color matrix C, and the adjacency matrix A of the graph, we define another integer matrix of dimensions $n \times r$, named as the *color distribution matrix* as follows:

$$\Phi = \mathcal{A}C, \tag{3}$$

where $\mathcal{A} = (A + I)$. Here, I is the identity matrix of dimensions $n \times n$.

The color distribution matrix gives information regarding the distribution of various colors within the network. In fact, it tells us about the exact number of different colors available in the closed neighborhood of any node.

Lemma 2.2.1 *Φ_{ij} is the number of nodes with the color j in the closed neighborhood of node v_i.*

Proof. Entries in the i^{th} row of matrix $\mathcal{A}$, denoted by $\mathcal{A}_i$, are 1 only for the vertices in $\mathcal{N}[v_i]$, and 0 otherwise. Entries in the j^{th} column of C, denoted by C_j, are 1 only for the vertices with the color j, and 0 otherwise. Thus, $\mathcal{A}_i C_j$ is the number of vertices that have color j in the closed neighborhood of vertex v_i. ∎

The color distribution matrix turns out to be a useful object in characterizing the distribution of colors to the vertices within a graph. For instance, it allows us to determine extra edges required to transform a given labeling of a graph into a completely heterogeneous one. In fact, $\Phi_{ij} = 0$ means that v_i is missing color j in its closed neighborhood. Thus, an extra edge is needed to connect v_i to some v_u with a color j. Upper and lower bounds on the number of extra edges required to get a completely heterogeneous graph with r labels from a given coloring of G are presented in the following result.

Theorem 2.2.1 *Let s be the maximum number of labels assigned to any vertex in a graph G. The number of extra edges $\mathcal{E}$, needed to get a completely heterogeneous graph with r labels from a given coloring of G is*

$$\left\lceil \frac{z(\Phi)}{2s} \right\rceil \leq \mathcal{E} \leq \ z(\Phi), \tag{4}$$

where $z(\Phi)$ is the number of 0's in the color distribution matrix Φ, for the given coloring.

Proof. Let $v_i \sim v_j$ be an extra edge connecting vertex v_i with colors $\kappa_1, \kappa_2, \cdots, \kappa_s$, to vertex v_j with colors $\tau_1, \tau_2, \cdots, \tau_s$. Since every vertex can have at most s distinct colors, $v_i \sim v_j$ can add at most s missing colors in $\mathcal{N}[v_i]$ and also at most s missing colors in $\mathcal{N}[v_j]$. This is possible whenever v_i is missing colors $\tau_1, \tau_2, \cdots, \tau_s$ in $\mathcal{N}[v_i]$ given by $\Phi_{i\tau} = 0,\ \forall \tau \in \{\tau_1, \cdots, \tau_s\}$, and v_j is missing $\kappa_1, \kappa_2, \cdots, \kappa_s$ in $\mathcal{N}[v_j]$, given by $\Phi_{j\kappa},\ \forall \kappa \in \{\kappa_1, \cdots, \kappa_s\}$. In this case, $v_i \sim v_j$ edge will change $2s$ zero entries in the Φ matrix to ones. In any other case, i.e., v_i has at least one of the $\tau_1, \tau_2, \cdots, \tau_s$

colors in its closed neighborhood or v_j has at least one of the $\kappa_1, \kappa_2, \cdots, \kappa_s$ colors in $\mathcal{N}[v_j]$, the number of zeros in Φ that will be converted to 1 will be less than $2s$. Thus, $\left\lceil \frac{z(\Phi)}{2s} \right\rceil \leq \mathcal{E}$.

The upper bound is straight forward as $\Phi_{i\tau} = 0$ means that v_i is missing the color τ in $\mathcal{N}[v_i]$, and the color τ can always be made available in $\mathcal{N}[v_i]$ through the addition of a single edge $v_i \sim v_j$, where v_j is any vertex with the color τ. ∎

As an illustration, consider G shown in Fig. 5.4. Every node has at most two labels from the set of five labels, given by $\{1, 2, 3, 4, 5\}$. The corresponding C and Φ matrices are,

$$C = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix}, \qquad \Phi = \begin{pmatrix} 1 & 1 & 1 & 1 & 2 \\ 2 & 2 & 1 & 1 & 2 \\ 1 & 1 & 2 & 2 & 2 \\ 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \end{pmatrix}.$$

Since $\Phi_{43} = \Phi_{51} = 0$, v_4 is missing label 3 in $\mathcal{N}[v_4]$ and v_5 is missing label 1 in its closed neighborhood. By adding $\mathcal{E}$ number of edges, where $1 \leq \mathcal{E} \leq 2$ (by Theorem 2.2.1), we can transform G into a completely heterogeneous graph. Note that by adding a single edge, $v_4 \sim v_5$, we get a completely heterogeneous graph with five labels.

2.2.1 Redundant Edges

In dynamic networks, edges may be lost. These edge deletions may take away certain resources from the neighborhood of an agent. Thus, we need to characterize edges whose deletion is not critical in the sense that their removal will preserve the number of resources available in the neighborhood of any agent. Let us define the *deficiency of a node* v as the number of colors from the coloring set $\{1, 2, \cdots, r\}$ that are missing in $\mathcal{N}[v_i]$. Similarly, *deficiency of a network* is the sum of all the node deficiencies.

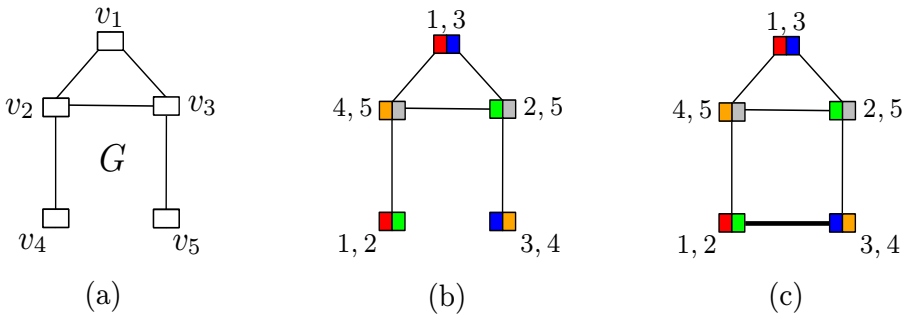


Figure 2.3: (a) A graph G has five vertices. (b) Two distinct labels are assigned to each vertex of G from the labeling set, $\{1, 2, \cdots, 5\}$. (c) As label 3 is missing from $\mathcal{N}[v_4]$, and label 1 is missing from $\mathcal{N}[v_5]$, $v_4 \sim v_5$ edge is needed to make G completely heterogeneous with five labels.

Based on this notion, we define a *redundant edge* to be the one whose deletion does not increase the deficiency of the network.

$\Phi_{ij} = 1$ means that v_i has only one neighbor with the color j. Thus, an edge between v_i and that j colored node is *not* redundant. Similarly, $\Phi_{ij} > 1$ implies that v_i has multiple nodes with the color j in $\mathcal{N}[v_i]$. As a result, there may be a redundant edge between v_i and some of its neighbors.

Theorem 2.2.2 *Let v_i be a node with colors $\kappa_1, \kappa_2, \cdots, \kappa_s$, and v_j be its neighbor with colors $\tau_1, \tau_2, \cdots, \tau_s$. An edge $v_i \sim v_j$ is redundant if and only if $\Phi_{i\tau_1}, \Phi_{i\tau_2}, \cdots, \Phi_{i\tau_s}$, and $\Phi_{j\kappa_1}, \Phi_{j\kappa_2}, \cdots, \Phi_{j\kappa_s}$ are all greater than 1 at the same time.*

Proof. ($\Rightarrow$) Let $v_i \sim v_j$ be a redundant edge. Then, by definition, it means that v_i has at least two neighbors for each of the colors $\tau_1, \tau_2, \cdots, \tau_s$ in $\mathcal{N}[v_i]$, i.e., $\Phi_{i\tau_1}, \Phi_{i\tau_2}, \cdots, \Phi_{i\tau_s}$ are all greater than 1. Similarly, for v_j , the redundancy of a $v_i \sim v_j$ edge implies that for each of the colors, $\kappa_1, \kappa_2, \cdots, \kappa_s$, vertex v_j has at least two neighbors in $\mathcal{N}[v_j]$, implying that $\Phi_{j\kappa_1}, \Phi_{j\kappa_2}, \cdots, \Phi_{j\kappa_s}$ are all greater than 1.

($\Leftarrow$) Assume $v_i \sim v_j$ is not redundant, then at least one of the following is true.

(a) there exists a $\tau \in \{\tau_1, \tau_2, \cdots, \tau_s\}$, such that v_i has only v_j as a τ colored vertex in $\mathcal{N}[v_i]$, i.e., $\Phi_{i\tau} = 1$ for some $\tau \in \{\tau_1, \tau_2, \cdots, \tau_s\}$. (b) there exists a $\kappa \in \{\kappa_1, \kappa_2, \cdots, \kappa_s\}$, such that v_j has only v_i as a κ colored vertex in $\mathcal{N}[v_j]$, i.e., $\Phi_{j\kappa} = 1$

for some $\kappa \in \{\kappa_1, \kappa_2, \cdots, \kappa_s\}$.

In both cases, $\Phi_{i\tau_1}, \Phi_{i\tau_2}, \cdots, \Phi_{i\tau_s}$ and $\Phi_{j\kappa_1}, \Phi_{j\kappa_2}, \cdots, \Phi_{j\kappa_s}$, are not all greater than 1 simultaneously, proving the required result. ∎

As an example, consider the graph in Fig. 2.4. Note that v_2 has labels 4, and 5, while v_3 has labels 2, and 5. In the color distribution matrix, $\Phi_{22}, \Phi_{25}, \Phi_{34}$, and Φ_{35} are all greater than 1. By Lemma 2.2.2, $v_2 \sim v_3$ edge is redundant and its deletion is not increasing the deficiency of any node in the network.

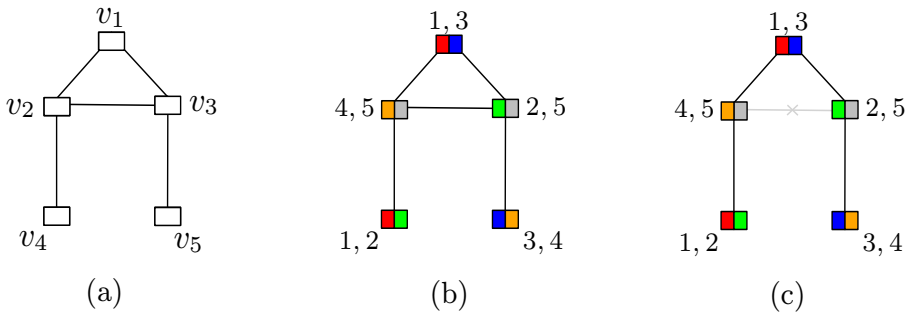


Figure 2.4: All the vertices in the graph are assigned two distinct labels from the set, $\{1, 2, \cdots, 5\}$. $v_2 \sim v_3$ edge is redundant. Removing this edge will not increase the deficiency of any node.

2.2.2 Most Deficient Color in the Network

The *most deficient color* in the network is the one that is missing from the closed neighborhood of maximum number of vertices in G. The j^{th} column of Φ tells about the availability of the color j in the closed neighborhood of all the vertices in G. By Lemma 2.2.1, $\Phi_{ij} = 0$ means v_i does not have a color j in $\mathcal{N}[v_i]$. Thus, the column index of Φ with the *maximum number of zeros* will be the most deficient color in the given labeling of G.

2.2.3 Effect of Node Deletion on the Labeling of Vertices

The deletion of a vertex from a graph may increase the deficiency of the remaining vertices. If vertex v_i with color κ is the only vertex with the color κ in the closed neighborhood of vertex v_j, deleting v_i will make v_j deficient in κ. However, if v_j

has more than one vertices with the color κ in $\mathcal{N}[v_j]$, which is indicated by $\Phi_{j\kappa} \geq 2$, removal of v_i will not increase the deficiency of v_j for the color κ. Using this observation, we write a matrix $\mathcal{U}$, in which $\mathcal{U}_{ij}$ is the number of vertices that will become deficient in color j upon the deletion of vertex v_i from the graph G. If C is the color matrix, and Φ be the color distribution matrix, then

$$\mathcal{U}_{i\kappa} = \begin{cases} \left|\{v_j : \quad v_j \in \mathcal{N}(v_i), \text{ and } \Phi_{j\kappa} \geq 2\}\right|, & \text{if } C_{i\kappa} = 1 \\ 0 & \text{otherwise.} \end{cases}$$

$\mathcal{U}$ is an integer matrix with dimensions $n \times r$, where n is the total number of vertices in G, and r is the total number of colors in the labeling of G. The i^{th} row sum of $\mathcal{U}$ indicates the increase in the deficiency of the network as a result of the deletion of v_i. If we define a *critical node* as the one whose removal from the network maximizes the increase in the deficiency of the remaining network, then the row index of $\mathcal{U}$ corresponding to the maximum row sum indicates the *most critical* vertex.

As an example, consider G shown in Fig. 2.5. The $\mathcal{U}$ matrix for G is,

$$\mathcal{U} = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 2 & 1 \\ 0 & 2 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

$\mathcal{U}_{13} = 1$, indicates that a single vertex will become deficient in color '3' upon the removal of v_1 from G. Note that both v_2 and v_3 are critical vertices here as both second and third rows have a maximum row sum.

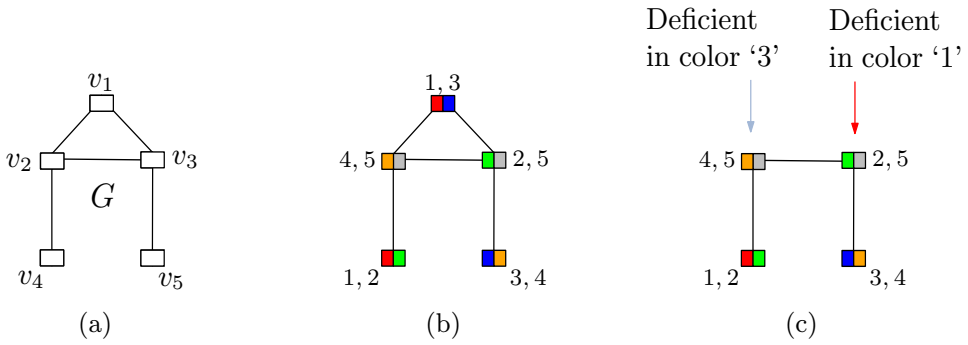


Figure 2.5: A graph G along with the labeling of its vertices from the set $\{1, 2, \cdots, 5\}$. Removal of v_1 makes v_2 deficient in color 3, while making v_3 deficient in color 1. Thus, deletion of v_1 will increase the deficiency of the network by two, which is also indicated by the row sum of the first row of $\mathcal{U}$.

2.3 Heterogeneity in Terms of the Maximum Number of Resources Available Within the Network

One of the primary characteristics of heterogeneous cooperative network is the maximum number of resources' types that can be incorporated within the system under the constraint that every node v can find every resource type in $\mathcal{N}[v]$. In other words, if r different types of resources are available within the network and each agent is equipped with at most s of these resources, then the maximum value of r, denoted by r^*, such that the the underlying graph of the network can be made completely heterogeneous with r^* labels is a crucial attribute of heterogeneous cooperative networks. In fact, in a completely heterogeneous network with r^* different types of resources, every agent finds all r^* types of resources in its closed neighborhood to accomplish various tasks. Thus, a higher value of r^* implies that more types of resources can be made available to the agents in a completely heterogeneous network. As a result, agents can perform tasks of higher complexity. It is to be noted that for a given graph G and a bound on the number of resources an agent can have, i.e., $\mid f(v) \mid \leq s$, if $r > r^*$, then it is impossible to distribute resources among nodes to get a completely heterogeneous graph with r labels. We utilize the notion of *domination in graphs* to address this issue. A *dominating set* is a fundamental object in the field of domination

in graphs.

As defined in Section 1.2, a set D is a *dominating set* if for each $v \in V$, either $v \in D$, or v is adjacent to some $u \in D$. It is to be noted that if D is a dominating set, then $\bigcup_{u \in D} \mathcal{N}[u] = V$. A graph can have multiple disjoint[1] dominating sets. A related and important concept is that of the *domatic number*, which is the maximum number of disjoint dominating sets in a graph. Thus, if D_i is a dominating set of a graph G and all the vertices in D_i are assigned a label i, i.e., $f(u) = i$, $\forall u \in D_i$, then, $i \in \mathcal{H}(v)$, $\forall v \in V$. In other words, label i is available in the closed neighborhood of every vertex in G. Furthermore, if the domatic number of a graph G is γ and every vertex is assigned a single label, then by the definition of the domatic number, G can be completely heterogeneous with at most γ labels, i.e., $r^* = \gamma$.

The notion of so-called (r, s)-configuration [66] defined below is also helpful in this context.

Definition 2.3.1 *Let* $\mathbf{r} = \{1, 2, \cdots, r\}$ *be a set of labels (colors). A function*

$$f : V \longrightarrow [\mathbf{r}]_s,$$

where $[\mathbf{r}]_s$ *is a collection of all* s*-subsets of* $\mathbf{r}$*, is called an* (r, s)*-configuration of a graph* G*, whenever* $\bigcup_{u \in \mathcal{N}[v]} f(u) = \mathbf{r}$, $\forall v \in V$.

Thus, the maximum value of r in an $(r, 1)$-configuration of a graph is the domatic number of the graph. Moreover, if γ is the domatic number of the graph, it is obvious that for $s > 1$, there always exist an (r, s)-configuration for $r = s\gamma$, i.e., each vertex in the graph can always be labeled with at most s colors such that the overall graph is completely heterogeneous with $s\gamma$ labels. However, there are graphs for which (r, s)-configurations exist for $r > s\gamma$. For example, cycle graphs C_n, in which n is not a multiple of 3, have a domatic number of 2, but $(5, 2)$-configurations of such graphs exist [66]. We present a sufficient condition for a graph to have an (r, s)-configuration

[1] i.e., intersection of distinct dominating sets is empty.

with $r = s\gamma + \lfloor \frac{s}{2} \rfloor$. A labeling scheme to obtain such a configuration can also be derived using this condition.

We begin by defining some terms that will be used to prove Theorem 2.3.1, which is the main result of this section.

Definition 2.3.2 *(Minimal Partition of G): Let G be a graph with domatic number γ, and vertex set V. A minimal partition of G, denoted by Π, is a partitioning of V into $\gamma + 1$ disjoint sets such that,*

$$\Pi = D_1 \cup D_2 \cup \cdots D_\gamma \cup V_\Pi, \tag{5}$$

where D_i is a minimal dominating set, $\forall i \in \{1, 2, \cdots, \gamma\}$, and $V_\Pi = V - (\cup_{i=1}^{\gamma} D_i)$ is the set of vertices that are not included in any minimal dominating set D_i. ∎

We term V_Π in (5) as the set of *non-critical vertices* with respect to the minimal partition Π, and we note that $V_\Pi \cap (\cup_{i=1}^{\gamma} D_i) = \emptyset$.

At the same time, consider a minimal partition of G, denoted by Π, and let $D_{\gamma+1}$ be a dominating set such that $V_\Pi \subseteq D_{\gamma+1}$. Since $dom(G) = \gamma$, and V_Π is not a dominating set, we have

$$D_{\gamma+1} = V_\Pi \cup I_\Pi,$$

where $I_\Pi \subset (\cup_{i=1}^{\gamma} D_i)$. We call a set I_Π with the smallest cardinality, a set of *common vertices* with respect to a minimal partition Π.

The notions of minimal partition Π, set of non-critical vertices with respect to Π, and set of common vertices with respect Π are shown in Fig. 2.6.

Theorem 2.3.1 *Let G be a graph with domatic number γ. Let Π be a minimal partition of G and I_Π be a set of common vertices with respect to Π. If there exists another minimal partition of G, say $\tilde{\Pi} \neq \Pi$, such that $I_\Pi \subseteq V_{\tilde{\Pi}}$, in which $V_{\tilde{\Pi}}$ is the set of non-critical vertices with respect to $\tilde{\Pi}$, then G has an (r, s)-configuration with $r = s\gamma + \lfloor \frac{s}{2} \rfloor$.*

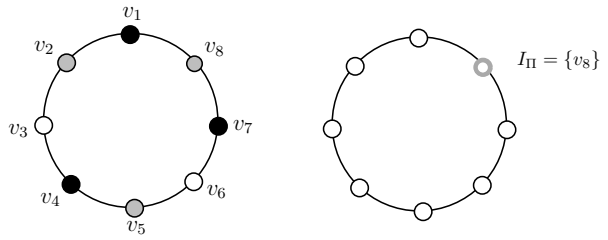


Figure 2.6: A cycle graph, C_8 having a domatic number $\gamma = 2$. A minimal partition $\Pi = D_1 \cup D_2 \cup V_\Pi$, where $D_1 = \{v_1, v_4, v_7\}$, and $D_2 = \{v_2, v_5, v_8\}$, are minimal dominating sets, and $V_\Pi = \{v_3, v_6\}$ is the set of non-critical vertices with respect to Π. We can take another dominating set D_3 as $D_3 = V_\Pi \cup I_\Pi$, where $I_\Pi = \{v_8\}$ is a set of common vertices with respect to a minimal partition Π.

Proof. Let $\Pi = \bigcup_{i=1}^{\gamma} D_i \cup V_\Pi$, in which V_Π is the set of non-critical vertices with respect to the minimal partition Π. Let $D_{\gamma+1}$ be a dominating set with $D_{\gamma+1} = V_\Pi \cup I_\Pi$, in which I_Π is a set of common vertices with respect to Π. Assign $\lfloor \frac{s}{2} \rfloor$ distinct labels to all the vertices in a dominating set D_i, for every $i \in \{1, 2, \cdots, \gamma + 1\}$[2]. Under this labelling scheme, the vertices in I_Π will have $\left(2 \lfloor \frac{s}{2} \rfloor\right)$ distinct labels as they are included in two different dominating sets, including $D_{\gamma+1}$ and some other D_i for $i \in \{1, 2, \cdots, \gamma\}$. Note that the vertices in I_Π are the only ones with $\left(2 \lfloor \frac{s}{2} \rfloor\right)$ labels, and every $v \in V$ has the set of $\lfloor \frac{s}{2} \rfloor (\gamma + 1)$ labels in its closed neighborhood.

Consider another minimal partition of G, $\tilde{\Pi} = \bigcup_{i=1}^{\gamma} S_i \cup V_{\tilde{\Pi}}$, with $V_{\tilde{\Pi}}$ being the set of non-critical vertices with respect to $\tilde{\Pi}$, and each S_i being a minimal dominating set. Let $\tilde{\Pi}$ be such that $I_\Pi \subseteq V_{\tilde{\Pi}}$. It means that every vertex in $V - V_{\tilde{\Pi}}$ has $\lfloor \frac{s}{2} \rfloor$ labels. Since $S_i \subseteq (V - V_{\tilde{\Pi}})$ for any $i \in \{1, 2, \cdots, \gamma\}$, every vertex $v \in S_i$ has $\lfloor \frac{s}{2} \rfloor$ labels. For every S_i, assign $\lceil \frac{s}{2} \rceil$ more unique labels to each vertex in S_i. Since each S_i is a dominating set, every $v \in V$ has a set of $\lceil \frac{s}{2} \rceil \gamma$ unique labels in $\mathcal{N}[v]$. Noting that $\lfloor \frac{s}{2} \rfloor (\gamma + 1)$ unique labels are already available in the closed neighborhood of every vertex, we get that all the vertices in V have now $\lfloor \frac{s}{2} \rfloor (\gamma + 1) + \lceil \frac{s}{2} \rceil \gamma = s\gamma + \lfloor \frac{s}{2} \rfloor$ distinct labels in their closed neighborhoods. Since each vertex is assigned at most s

[2] $\lfloor \frac{s}{2} \rfloor$ labels assigned to the vertices of D_i are different from the ones assigned to the vertices in D_j where $i \neq j$.

distinct labels, we have an (r, s)-configuration of G with $r = s\gamma + \lfloor \frac{s}{2} \rfloor$. ∎

As an example, consider a $(5, 2)$-configuration of C_8. Domatic number of C_8 is 2, i.e., $\gamma = 2$. We consider two minimal partitions of C_8, denoted by Π and $\tilde{\Pi}$, where Π is shown in Fig. 2.6. For $\tilde{\Pi}$, we take $\tilde{\Pi} = S_1 \cup S_2 \cup V_{\tilde{\Pi}}$, as shown in Fig. 2.7. Since $I_\Pi \subseteq V_{\tilde{\Pi}}$, $(5, 2)$-configuration exists for C_8.

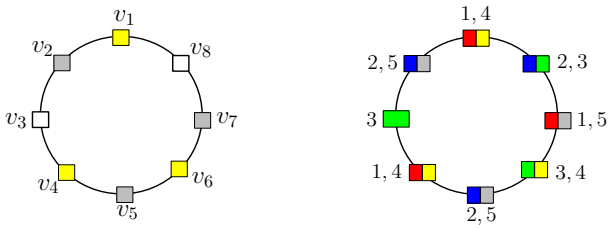


Figure 2.7: (a) $\tilde{\Pi} = S_1 \cup S_2 \cup V_{\tilde{\Pi}}$, in which $S_1 = \{v_1, v_4, v_6\}$, and $S_2 = \{v_2, v_5, v_7\}$ are disjoint minimal dominating sets, and $V_{\tilde{\Pi}} = \{v_3, v_8\}$ is the set of non-critical vertices with respect to $\tilde{\Pi}$. (b) A $(5, 2)$-configuration of C_8.

2.3.1 Assignment of Miltiple Resources in R-Disk Proximity Graphs

The R-disk proximity graph model is frequently employed to model inter-connections among nodes in multiagent networks. In such a model, a disk of radius R, which represents the interaction range of a node, is associated with every node v that lies at the center of the disk. A node forms an edge with other nodes if and only if they exist within the R radius disk of the node. Applications of such a model include ad hoc communication networks, wireless sensor networks (e.g., see [98]), multiagent and multirobot systems (see e.g., [99]), and other broadcast networks with a limited range transmitters and receivers, to name a few.

Analysis of (r, s)-configurations of R-disk proximity graphs is of significance, particularly in the context of heterogeneous multiagent systems. In this section, we show that for a given s, R-disk graphs always have an (r, s)-configuration for $r = \lfloor \frac{5s}{2} \rfloor$ under some mild conditions. It is assumed that the agents equipped with multiple capabilities or resources are lying in a plane, and interactions among them are modeled by the R-disk proximity graph.

We start by translating the geometric property of such graphs into a graph-theoretic one by first defining the following special graphs. A graph G is a *complete bi-partite graph* if there exists a partition of its vertex set, $V = X \cup Y$, such that an edge $u \sim v$ exists whenever $u \in X$ and $v \in Y$. If $|X| = x$ and $|Y| = y$, then a complete bi-partite graph is denoted by $K_{x,y}$. Examples of complete bi-partite graphs are shown in Fig. 2.8. We also define a *double cycle graph*, denoted by $C_4 \bullet C_4$, as the one obtained by identifying a vertex of C_4 with a vertex of another C_4, as shown in Fig. 2.8. Furthermore,a graph G is said to be an *H-free graph*, if H is not an induced sub-graph of G.

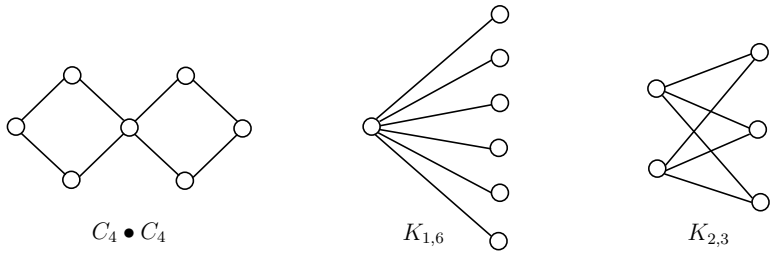


Figure 2.8: The double cycle graph, $C_4 \bullet C_4$, obtained by identifying a vertex of C_4 with a vertex of another C_4, and complete bi-partite graphs, $K_{1,6}$, and $K_{2,3}$.

It is shown in [100] that $K_{2,3}$ cannot be an R-disk graph. In the following Lemma, it is shown that R-disk graphs are always $K_{1,6}$-free.

Lemma 2.3.1 *An R-disk proximity graph is $K_{1,6}$ free.*

Proof. Let $G(V, E)$ be an R-disk proximity graph. Let $v \in V$, such that $\mathcal{N}(v) = \{v_1, v_2, \cdots, v_p\}$, where $p \geq 6$. We define $\theta_{(v_i v v_j)}$ to be the angle v makes with v_i and v_j. If $\|v_i, v_j\|$ is the euclidean distance between v_i and v_j, then it is easy to see that $\|v_i, v_j\| > R$, whenever $\theta_{(v_i v v_j)} > 60^o$. Thus, $v_i, v_j \in \mathcal{N}(v)$ are non-adjacent if and only if $\theta_{(v_i v v_j)} > 60^o$. To have $K_{1,6}$ as an induced sub-graph of G, there must be a subset, $\tilde{\mathcal{N}}(v) \subseteq \mathcal{N}(v)$ with $|\ \tilde{\mathcal{N}}(v)\ | = q \geq 6$, such that $\theta_{(x_i v x_j)} > 60^o$, $\forall x_i, x_j \in \tilde{\mathcal{N}}(v)$. This will give $\sum_{i=1}^{q-1} \theta_{x_i v x_{(i+1)}} + \theta_{x_q v x_1} > 360^o$, which is not possible. Thus, an R-disk graph is $K_{1,6}$-free. ∎

Since every R-disk graph is $K_{1,6}$-free, so now we can focus on the results regarding (r, s)-configurations of $K_{1,6}$-free graph to study the resource assignment problem in R-disk graphs. An important result regarding (r, s)-configurations of $K_{1,6}$-free graphs is stated below [26].

Theorem 2.3.2 *For any positive integer s, a connected $K_{1,6}$-free graph G with a minimum degree of at least two has an (r, s)-configuration with $r = \lfloor \frac{5s}{2} \rfloor$ whenever G is not isomorphic to $C_4, C_7, K_{2,3}$, or $C_4 \bullet C_4$.*

Proof. See Appendix A

Now, using Theorem 2.3.2, Lemma 2.3.1, and the fact that an R-disk graph can never have a component isomorphic to $K_{2,3}$ graph, we get the following result regarding (r, s)-configurations of R-disk graphs.

Theorem 2.3.3 *For any positive integer s, an R-disk proximity graph G with a minimum degree of at least 2 has an (r, s)-configuration with $r = \lfloor \frac{5s}{2} \rfloor$ whenever G has no component isomorphic to C_4, C_7, or $C_4 \bullet C_4$.*

Example:

Consider a group of robots deployed in a certain region $\mathcal{D} \subset \mathbb{R}^2$ for the purpose of environment modeling of the location. The robots interact and exchange information with each other, and this interaction is modeled by an $R - disk$ proximity graph model in which every robot v coordinates (form an edge in the underlying graph of the network) with all robots that lie within a (euclidean) distance R from v.

For the purpose of environment modeling of $\mathcal{D}$, we consider five different environment parameters including temperature, relative humidity, air pressure, light, and soil texture that are observed through various types of sensors. Each robot performs some spatio-temporal data processing using the data of the five environmental parameters obtained from the sensors that are mounted on robots. If each sensor observes a specific environmental parameter, five different types of sensors are needed.

One possible course of action is to install all five sensors on each robot so that robots can collect information of all parameters for further processing by themselves. However, this might not be a feasible approach for one or more of the following reasons: first a large amount of power supplies will be needed to keep all five sensors operational, and power is always a limiting factor for a continuous operation of such networks over an extended period of time; second a large number of sensors of each type will be required which may not be cost effective; third mounting all sensors on a single robot may not be feasible from hardware view-point in certain cases. Another approach is to install a subset of sensors on each robot and utilize the fact that robots exchange information with each other. But, this approach requires sensors to be distributed among robots in such a way that each robot can obtain the data of the missing parameters from its neighbor robots. In other words, sensors missing on a robot are installed on the neighbor robots. This set-up requires a much smaller number of sensors of each type for the overall operation of the system, thus, reducing the overall energy consumption. For instance, if each robot is allowed to have at most two of the five sensors' types, then sensors need to be installed in such a way that each robot can find a complete set of five distinct sensors in its closed neighborhood. However, it is possible if and only if the underlying R-dsik graph of the robot network has a $(5, 2)$-configuration. It is shown in Theorem 2.3.3 that every R-disk graph has a $(5, 2)$-configuration under some conditions. Thus, it is possible to make a robot network completely heterogeneous with five labels under the restriction that each robot can have at most two labels. An example is illustrated in Fig. 2.9.

2.3.2 Assignment of Miltiple Resources in Cycle Graphs

Here, we analyze the maximum number of resources that can be made available in the closed neighborhood of a node in a cycle graph under the constraint on the maximum number of resources contained by a node itself. In other words, for a given s, what

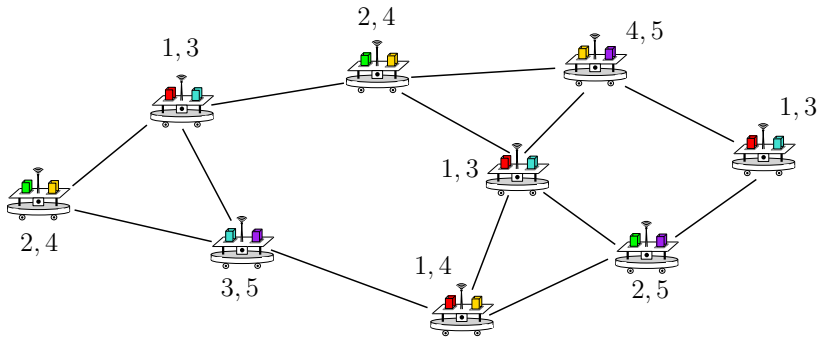


Figure 2.9: A group of robots connected via R-disk proximity graph model. Each robot is assigned two labels from the set $\{1, 2, 3, 4, 5\}$ in such a way that a complete set of five labels is available in the closed neighborhood of every node.

is the maximum r (i.e., r^*) possible in an (r, s)-configuration of a cycle graph. From Theorem 2.3.2, r^* is at least $\left\lfloor \frac{5s}{2} \right\rfloor$. Can we do better than that? The answer is yes.

Theorem 2.3.4 *Let C_n be a cycle graph with $n = 3m + x$ vertices, where m is any positive integer and $x \in \{0, 1, 2\}$. For a given positive integer s, the maximum value of r in an (r, s)-configuration of C_n is*

$$r_{\max} = \begin{cases} 3s & \text{if } n = 3m \\ 3s - 1 & \text{if } n = 3m + 2, \quad s \leq (m + 1) \\ 3s - 2 & \text{if } n = 3m + 1, \quad \frac{m+1}{2} < s \leq (m + 1) \end{cases}$$

Proof. Let C_n be a cycle graph with n vertices. We assume that the vertices in cycle C_n are arranged in the order $v_1, v_2, \cdots, v_n$, and vertices v_n and v_1 are connected by an edge. Further, we consider three cases of cycle graphs based on n.

Case 1 $[n = 3m]$*:* m is any positive integer. The domatic number of such a cycle graph is 3, i.e., there are three disjoint dominating sets. Pick a dominating set and assign s distinct labels to all of its vertices. Repeat the same for the other two dominating sets to get a $(3s, s)$-configuration.

Case 2 $[n = 3m + 2]$*:* We observe that $3m+2$ vertices can accommodate $s(3m+2)$ labels altogether. The size of any dominating set is at least $m+1$, which implies that

each label must be assigned to at least $(m+1)$ vertices. Thus, an (r, s)-configuration is possible for r that is at most $\left\lfloor \frac{s(3m+2)}{m+1} \right\rfloor$.

$$\left\lfloor \frac{s(3m+2)}{m+1} \right\rfloor \;=\; \left\lfloor 2s \;+\; \frac{sm}{m+1} \right\rfloor \;=\; 2s \;+\; \left\lfloor \frac{sm}{m+1} \right\rfloor$$

Using Hermite's identity, we get

$$2s \;+\; \left\lfloor \frac{sm}{m+1} \right\rfloor \;=\; 2s \;+\; \sum_{k=1}^{s-1} \left\lfloor \frac{m}{m+1} \;+\; \frac{k}{s} \right\rfloor$$

$\left\lfloor \frac{m}{m+1} \;+\; \frac{k}{s} \right\rfloor = 1$ whenever $k \geq \frac{s}{m+1}$, which is true for every $k \geq 1$ under our assumption of $s \leq m+1$. Thus, we get

$$2s \;+\; \sum_{k=1}^{s-1} \left\lfloor \frac{m}{m+1} \;+\; \frac{k}{s} \right\rfloor \;=\; 2s \;+\; (s-1) \;=\; 3s \;-\; 1$$

To obtain a $(3s-1, s)$ configuration of C_{3m+2}, consider the table shown in Fig. 2.10, and assign each vertex v_i a set of labels in the corresponding column of the table. Note that there are exactly s number of labels in each column[3]. Moreover, union of labels of any three consecutive columns is the set $\{1, 2, \cdots, 3s-1\}$. Thus, for a given $s \leq m+1$, we get an (r, s)-configuration for $r = 3s-1$.

Case 3 $[n = 3m+1]$*:* In this case, $3m+1$ vertices can pack $s(3m+1)$ labels altogether. In any dominating set, there will be at least $(m+1)$ vertices. Therefore, at least $(m+1)$ vertices should contain any single label to have an (r, s)-configuration. This implies that $r_{\max} \leq \left\lfloor \frac{s(3m+1)}{m+1} \right\rfloor$.

$$\left\lfloor \frac{s(3m+1)}{m+1} \right\rfloor \;=\; \left\lfloor s \;+\; \frac{2sm}{m+1} \right\rfloor \;=\; s \;+\; \left\lfloor \frac{2sm}{m+1} \right\rfloor \;=\; s \;+\; \sum_{k=0}^{2s-1} \left\lfloor \frac{m}{m+1} \;+\; \frac{k}{2s} \right\rfloor$$

The term $\left\lfloor \frac{m}{m+1} \;+\; \frac{k}{2s} \right\rfloor = 1$, when $\frac{m+1}{2} < s \leq (m+1)$, and $k \geq 2$. Thus,

$$s \;+\; \sum_{k=0}^{2s-1} \left\lfloor \frac{m}{m+1} \;+\; \frac{k}{2s} \right\rfloor \;=\; s \;+\; 2s \;-\; 2 \;=\; 3s \;-2$$

[3]There is exactly one label in each of the s rows of every column except for the columns $v_{(3i+2)},\ i \in \{1, 2, \cdots, m\}$. In each of the column $v_{3i+2},\ i \in \{1, 2, \cdots, m\}$, there is only one row that contains two labels. However, the s^{th} row in all such columns is empty. Thus, every column $v_{(3i+2)},\ i \in \{1, 2, \cdots, m\}$ also has exactly s number of labels.

A labeling scheme to obtain an (r, s)-configuration of C_{3m+1} with $\frac{m+1}{2} < s \leq (m+1)$ and $r = 3s-2$ is given the table in Fig. 2.11. Every vertex v_i is assigned labels in the v_i column. Each column has s labels, and union of labels in any three consecutive columns is the set $\{1, 2, \cdots, 3s-2\}$, thus an (r, s)-configuration is obtained with the desired s and r. ∎

	v_1	v_2	v_3	v_4	v_5	v_6	$\cdots$	v_{3m-5}	v_{3m-4}	v_{3m-3}	v_{3m-2}	v_{3m-1}	v_{3m}	v_{3m+1}	v_{3m+2}
1^{st} row	1	2	3	1	2	3	$\cdots$	1	2	3	1	2	3	1	2, 3
2^{nd} row	4	5	6	4	5	6	$\cdots$	4	5	6	4	5, 6	4	5	6
3^{rd} row	7	8	9	7	8	9	$\cdots$	7	8, 9	7	8	9	7	8	9
	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$(s-1)^{st}$row	$(3s-5)$	$(3s-4)$	$(3s-3)$	$(3s-5)$	$(3s-4)$, $(3s-3)$	$(3s-5)$	$\cdots$	$(3s-4)$	$(3s-3)$	$(3s-5)$	$(3s-4)$	$(3s-3)$	$(3s-5)$	$(3s-4)$	$(3s-3)$
s^{th} row	$(3s-2)$	$(3s-1)$	$(3s-2)$	$(3s-1)$	$\times$	$(3s-2)$	$\cdots$	$(3s-1)$	$\times$	$(3s-2)$	$(3s-1)$	$\times$	$(3s-2)$	$(3s-1)$	$\times$

Figure 2.10: Labeling scheme for an (r, s)-configuration of C_{3m+2} for $r = 3s-1$.

	v_1	v_2	v_3	v_4	v_5	v_6	$\cdots$	v_{3m-5}	v_{3m-4}	v_{3m-3}	v_{3m-2}	v_{3m-1}	v_{3m}	v_{3m+1}
1^{st} row	1, 3	2	3	1	2	3	$\cdots$	1	2	3	1	2	3	1, 2
2^{nd} row	4	5	6	4	5	6	$\cdots$	4	5	6	4, 5	4, 6	5	6
3^{rd} row	7	8	9	7	8	9	$\cdots$	7, 8	7, 9	8	9	7	8	9
	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$(s-1)^{st}$row	$(3s-5)$	$(3s-4)$	$(3s-3)$	$(3s-5)$, $(3s-4)$	$(3s-5)$, $(3s-3)$	$(3s-4)$	$\cdots$	$(3s-3)$	$(3s-5)$	$(3s-4)$	$(3s-3)$	$(3s-5)$	$(3s-4)$	$(3s-3)$
s^{th} row	$\times$	$(3s-2)$	$(3s-2)$	$\times$	$\times$	$(3s-2)$	$\cdots$	$\times$	$\times$	$(3s-2)$	$\times$	$\times$	$(3s-2)$	$\times$

Figure 2.11: Labeling scheme for an (r, s)-configuration of C_{3m+1} for $r = 3s-2$.

Examples illustrating the labeling schemes described above are shown in Fig. 2.12 for C_{10} and C_{11}.

2.4 Discussion and Generalizations

In Section 2.1, we introduced the notion of a completely heterogeneous graph with a set of r unique labels as the one in which every node finds a complete set of r labels in its closed neighborhood. There might be situations in which a complete

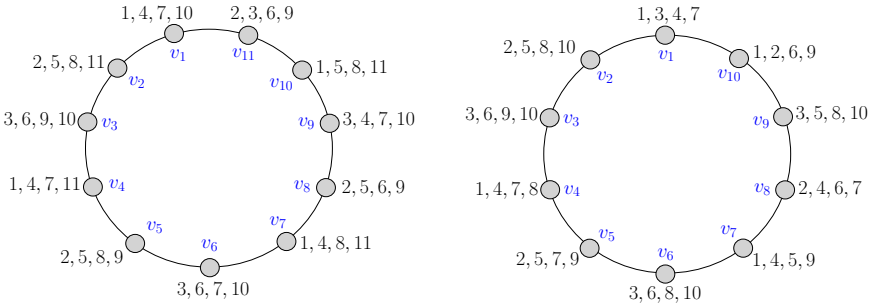


Figure 2.12: $(4, 11)$-configuration and $(4, 10)$-configuration of C_{11} and C_{10} respectively.

set of resources might not be required in the closed neighborhood of all nodes. This motivates to further extend the notion of a completely heterogeneous graph towards more general scenarios. For instance, the notion of neighborhood can be extended to the *k-neighborhood*. If the distance between nodes v and u, denoted by $d(u, v)$, is the length of the shortest path in a graph G, then the *open k-neighborhood* of node v is the set $\mathcal{N}_k(v) = \{u \in V : \ d(u, v) \leq k\}$. Likewise, the *closed k-neighborhood* of v is $\mathcal{N}_k[v] = \mathcal{N}_k(v) \cup v$. We can define a *completely k-heterogeneous graph* with r labels as the one in which every vertex finds a complete set of r labels in its closed k-neighborhood. The concept of completely k-heterogeneous graph with r labels is particularly useful in the situations where r is quite large, i.e., a large number of different types of resources exist within the network, and it might not be possible to ensure the availability of all the resources in the closed neighborhood of every node. In such situations, one can aim to distribute resources among the nodes to get a completely k-heterogeneous graph with r labels for a smaller value of k to ensure that each node finds all r resources within a small distance k from it.

Since the color distribution matrix, Φ, has been a fundamental object for the analysis in Section 2.2, we can define the k-color distribution matrix as follows for the case of completely k-heterogeneous graphs.

$$\Phi(k) = \mathcal{A}_k C$$

where $\mathcal{A}_k = A_k + I$ and C is a color matrix. A_k is an $n \times n$ matrix whose ij^{th} entry is 1 whenever $d(v_i, v_j) \leq k$ and $i \neq j$.

It is to be observed that ij^{th} entry of the $\Phi(k)$ matrix is the number of vertices with a color j in the closed k-neighborhood of v_i. Thus, we can use the same approach as in Section 2.2 to analyze the distribution of labels when using the notion of k-neighborhood for some $k > 1$.

CHAPTER III

EXPLOITING HETEROGENEITY IN WIRELESS SENSOR & ACTOR NETWORKS

In Chapter 2, a network topology based characterization of heterogeneity in multiagent systems is presented. The underlying objective is to maximally utilize the available resources within the network through local interactions. In this chapter, the issue of energy-efficient data collection in heterogeneous wireless sensor and actor networks (WSANs) is addressed using the methods introduced in Chapter 2. Heterogeneous WSANs provide effective solutions to several distributed sensing and response related problems. In such networks, managing the activity of sensor nodes through efficient activity scheduling mechanisms is imperative for a longer lifetime of the network because of the limited power resources. We demonstrate that using our framework, interactions between actors, and heterogeneity among sensors can be utilized to design an energy-efficient scheduling scheme while ensuring that even after the deactivation of a certain number of sensors, actors continue to obtain the same information as they were acquiring when all sensors were on.

The components in WSANs are categorized into two major classes: sensors and actors. Sensors provide a distributed sensing infrastructure, and are typically inexpensive, low-power devices with limited computational and communication capabilities [69]. Owing to these properties, sensors are generally deployed in greater numbers. Actors, on the other hand are more sophisticated and resource-rich nodes with longer battery life, higher processing skills, and transmission powers. They are capable of processing data obtained from the sensors, and then taking appropriate actions. Robots and unmanned ground or aerial vehicles are examples of actors. Typically it

is assumed that the number of actors in a network is much smaller than the sensors [70].

The rest of the chapter is organized as follows: Section 3.1 provides a description of the heterogeneous wireless and actor network along with the objective of our scheme; in Section 3.2, a graph-theoretic formulation of the problem is provided; an energy-efficient scheduling scheme for a longer lifetime of the WSAN is presented in Section 3.3; analysis of the random distribution of sensor and actor nodes is performed in Section 3.4 along with an illustration of the scheme in Section 3.5.

3.1 System Description

Let there be r different types of sensors. Sensors of each type are deployed at random in some domain $\mathcal{A} \subset \mathbb{R}^2$ such that the location of each sensor is independent of other sensors' locations. Such a deployment of sensors can be modeled as a stationary Poisson point process with constant intensity[1] [108]. All of the sensors have footprints of the form of closed balls of some radii that depend on the types of sensors. Let us say that each sensor belongs to one of the types in the set $\mathbf{r} = \{1, 2, \cdots, r\}$, then the deployment of sensors of each type can be modeled by a stationary Poisson point process with intensity λ_i where $i \in \mathbf{r}$. Further, we use the following notations:

r : total number of sensor types.

λ_i : expected number of sensors of type i per unit area modeled as a stationary Poisson point process.

Δ_i : radius of the footprint of a sensor of type i.

α_i : area of the sensor footprint of type i ($\alpha_i = \pi\Delta_i^2$).

Meanwhile, actors (robots), which are the resourceful nodes within the network and are capable of performing different tasks after receiving data from various sensors,

[1]Expected number of sensors in a unit area.

are also distributed at random and independent of other actors' locations. Thus, actors can also be modeled by a stationary Poisson point process with intensity ρ. As in the automated architecture of WSANs, actors coordinate with each other by communicating and exchanging information. An actor interacts with all other actors lying within the distance Δ_a from it. This gives an interaction network that can be modeled by R-disk proximity graphs, in which $R = \Delta_a$. We will use the following notation throughout this chapter.

ρ : expected number of actors per unit area.

Δ_a : communication range of an actor

(radius of the footprint of an actor).

α_a : area of the actor's foot print ($\alpha_a = \pi\Delta_a^2$).

As mentioned earlier, typically the number of actors is much smaller than the number of sensors, thus, $\rho < \lambda_i,\ \forall i \in \mathbf{r}$. Moreover, actors have higher transmission ranges, i.e., Δ_a is usually higher than Δ_i. A sensor which is in an active mode (on state) transmits its data to actors lying within its footprint. Sensors do not communicate with each other, whereas actors transmit and receive information from other actors as well as sensors. Every actor performs tasks that require data from every sensor type, i.e., an actor needs to have information of all r sensing parameters. We consider that the spatial gradients of the sensing parameters observed by sensors are not too large within the field of observation, i.e., there are no abrupt variations in the sensing parameters throughout the field of interest. Therefore, an actor can receive information regarding the i^{th} ($i \in \mathbf{r}$) sensing parameter either directly from the sensor if there exits a sensor of type i within Δ_i distance from the actor, or through one of the neighboring actors which is directly receiving data from a sensor of type i.

Under this set-up, the probability of an actor to receive information regarding all different sensing parameters in $\mathbf{r}$, either directly from sensors, or through adjacent actors depends on various factors including λ_i and Δ_i. Thus, for each $i \in \mathbf{r}$, increasing

λ_i (number of sensors of type i) and Δ_i (transmission range of a sensor of type i), will increase the number of actors receiving information of all r different sensing parameters, but only at an additional cost. However, it is observed that owing to the random deployment of sensors with λ_i intensities, there exist redundancy within the network in the sense that a lot more sensors are on than required. We can get rid of this redundancy by turning off the redundant sensors for an energy-efficient operation of the system.

Thus, our *objective* is to develop a systematic scheme to turn off the maximal number of redundant sensors of all types in a distributed manner while ensuring the following: if an actor v, or one of its neighbor actors (actors directly connected to v) are lying within the footprint of a sensor of type i in the initial deployment (when all sensors are on), then the same should be true even when the redundant sensors are turned off.

3.2 A Graph-Theoretic Model

The problem under consideration can be investigated in graph-theoretic terms. The network of actors (robots) can be modeled by a graph $G(V, E)$, in which the vertex set V represent actors and the edges in the edge set E correspond to interactions among them. Heterogeneity that exists within the system in the form of sensors of various types can be modeled using the *graph coloring* notion introduced in Section 2.1.

Here, vertices in the graph (representing an actor network) are colored in accordance to the types of sensors directly transmitting data to the corresponding actors. Since r different types of sensors are available within the system, the coloring set is $\mathbf{r} = \{1, 2, \cdots, r\}$. Vertices in the graph G are then assigned labels according to the following labeling function:

$$f : V \longrightarrow 2^{\mathbf{r}}$$

where $2^{\mathbf{r}}$ is the set of all subsets of r.

If there exists at least one sensor of type $i \in \mathbf{r}$ within a distance Δ_i from an actor v, then the corresponding vertex in G will be assigned the label (color) i. Thus,

$$f(v) = \left\{ i \in \mathbf{r} \mid \begin{array}{l} \text{at least one sensor of type } i \text{ exists} \\ \text{within } \Delta_i \text{ distance from } v. \end{array} \right\} \tag{6}$$

An actor receiving data directly from a sensor of type i exchanges it with the neighboring actors as actors are interacting and exchanging information with each other. Further, we define $\mathcal{H}(v) \subseteq \mathbf{r}$ as the set of colors that a vertex v can find in its closed neighborhood, i.e.,

$$\mathcal{H}(v) = \bigcup_{u \in \mathcal{N}[v]} f(u)$$

Vertices in the graph G are actors and are labeled in accordance with the types of sensors that directly transmit data to actors. For instance, actor u lies within the footprints of sensors of type 1 and 2, the corresponding vertex u in G is assigned labels 1 and 2, i.e., $f(u) = \{1, 2\}$. Also, u is directly connected to v and x which have labels 3 and 2 respectively, thus, $\mathcal{H}(u) = \mathbf{r} = \{1, 2, 3\}$.

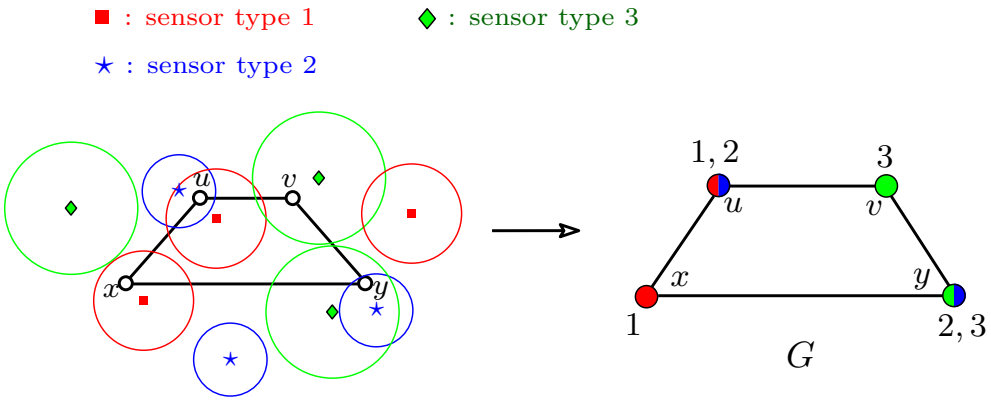


Figure 3.1: Three different types of sensors are distributed. The vertices in the graph G (representing an actor network) are assigned colors (labels) from the set $\mathbf{r} = \{1, 2, 3\}$ in accordance with the sensor types directly available to the corresponding actors.

3.2.1 Objective

Sensors of each type $i \in \mathbf{r}$ are distributed at random and independent of each other with intensity λ_i. Thus, colors (labels) assigned to vertices in the above graph-theoretic model directly depend on the distribution of sensors. We call the labeling of vertices in G due to the initial random deployment of sensors as $\mathcal{L}_{ini}(G)$. Under the labeling $\mathcal{L}_{ini}(G)$, a vertex v in G is assigned labels $f_{ini}(v)$, and $\mathcal{H}_{ini}(v)$ is the set of labels available in the closed neighborhood of v. Thus, our goal is to develop a systematic scheme to obtain a new labeling of G, i.e., $\mathcal{L}_{new}(G)$ from $\mathcal{L}_{ini}(G)$ by getting rid of some of the labels (redundant labels) assigned to the vertices while ensuring that under this new labeling (which is derived from $\mathcal{L}_{ini}(G)$), every vertex finds exactly the same set of labels in its closed neighborhood as in $\mathcal{L}_{ini}(G)$. More precisely, for every vertex v in G, we want to find $f_{new}(v) \subseteq f_{ini}(v)$ in a distributed manner such that $\mathcal{H}_{new}(v) = \mathcal{H}_{ini}(v)$. In a special case where $\mathcal{H}_{ini} = \mathbf{r}$, we get a completely heterogeneous graph. Since labels assigned to vertices directly correspond to the sensors transmitting data to the actors, getting rid of the labels mean that the corresponding sensors can be turned off leading towards an energy-efficient operation of the wireless sensor and actor network.

3.3 Energy-Efficient Data Collection Scheme

In this section, we present a scheme to turn off redundant sensors for energy efficiency. Our proposed scheme utilizes both randomness in the deployment of sensors of various types within the region $\mathcal{A} \subset \mathbb{R}^2$, and coordination among actors to determine and resolve the redundancy existing within the sensor network. Every sensor is considered to have two modes, *active* (on) mode and *de-active* (off) mode. A sensor transmits its data to the actors lying within sensor's footprint only in the active mode. Our scheme consists of the following rounds:

3.3.1 Randomization

Sensors of each type $i \in \mathbf{r}$ are deployed randomly and independently of each other with intensity λ_i'. At time $t = 0$, each sensor enters into the active mode with some probability $p > 0$. Thus, the effective intensity of sensors of type i will be $\lambda_i = p\lambda_i'$. In order to keep the expected on time same for all sensors during the overall lifetime of the network, this step is repeated after some fixed interval t_δ.[2] Sensors that become active as a result of this step start transmitting their data to the actors lying within the footprints of these sensors. Every actor v maintains a list of the types of sensors that are directly transmitting data to the actor, i.e., $f(v)$ as defined in (6).

3.3.2 Determination of Redundant Sensors

Once $f(v)$ is determined by every actor v, the next key step is the exchange of $f(v)$ by every v with its neighbors to determine the existence of redundant sensors within the footprint of an actor. Once determined, these redundant sensors will be de-activated through a de-activating message by the actor to the sensors. The graph-theoretic model of the system introduced in Section 3.2 will be used here for the purpose of determining redundant sensors. Every vertex in a graph G modeling the actor network is assigned labels $f(v)$.

Our goal is to obtain for every $v \in V$, a subset $s(v) \subseteq f(v)$ with the minimum cardinality such that $\bigcup_{u \in \mathcal{N}[v]} s(u) = \mathcal{H}(v)$. We deal with this problem individually for each $i \in \mathbf{r}$. Let V_i be the set of vertices having label i, i.e.,

$$V_i = \{v \in V : \ i \in f(v)\}$$

Also, let $\tilde{V}_i$ be the set of vertices that have at least one vertex with a label i in their closed neighborhoods, i.e.,

$$\tilde{V}_i = \{v \in V : \ i \in \mathcal{H}(v)\}$$

[2] i.e., after every t_δ interval, a sensor enters into an active mode with some probability p.

It is to be noted that $\tilde{V}_i = \bigcup_{v \in V_i} \mathcal{N}[v]$. Thus, for the label i, we need to find a subset $S_i \subseteq V_i$ with the minimum cardinality such that $\bigcup_{s \in S_i} \mathcal{N}[s] = \tilde{V}_i$.

In a special case when every vertex is assigned the label i, i.e., $V_i = V$, this problem becomes a *dominating set* problem. For our case, when V_i may not be equal to V, we define the *restricted dominating set* as

Definition 3.3.1 *(Restricted Dominating Set) Let $V_i \subseteq V$, a subset $S_i \subseteq V_i$ is a restricted dominating set with respect to V_i whenever $\bigcup_{s \in S_i} \mathcal{N}[s] = \bigcup_{v \in V_i} \mathcal{N}[v]$.*

An example is shown in Fig. 3.2.

(a) (b)

Figure 3.2: (a) The circled vertices form a dominating set of the graph. (b) If $V_1 = \{v_1, v_2, v_3, v_4\}$, then a restricted dominating set with respect to V_1 consist of the circled vertices, i.e., $S_1 = \{v_1, v_2, v_4\}$.

Computation of a Restricted Dominating Set S_i:

The problem of finding a minimum dominating set is NP-hard (e.g., [60]) leading to the fact that finding a minimum restricted dominating set is also NP-hard. Thus, finding efficient algorithms for the approximate solutions has been an active area of research. The simplest approach is the *greedy approach* in which a vertex covering the maximum number of uncovered vertices is added into a dominating set at each step [109]. The greedy algorithm achieves an approximation ratio of $\ln \Lambda$ in $O(n)$ time, where Λ and n are the maximum degree[3] and total number of vertices in the graph respectively [109]. A *distributed* version of the greedy algorithm is presented in [110], [111]. Interestingly, it is shown in [60] that unless P≈NP, the $\ln \Lambda$-approximation ratio of the simple greedy approach is optimal (upto lower order terms). Therefore,

[3]degree of a vertex v is the cardinality of $\mathcal{N}(v)$.

the problem of finding a small *restricted dominating set* with respect to $V_i \subseteq V$ can be solved using the simplest distributed greedy approach. Below, we present a distributed greedy algorithm adapted from [110] for finding a restricted dominating set with respect to V_i. Unlike [110], in which every $v \in V$ executes a greedy routine, here the algorithm is executed only by the vertices in V_i.

Let us define a *dominated node* as the one whose closed neighborhood contains at least one vertex form the restricted dominating set. A vertex is said to be *undominated* if it is not a dominated one. Also let $\mathcal{U}(v) =$ number of undominated nodes in $\mathcal{N}[v]$.

Algorithm I: Restricting Dominating Set w.r.t $V_i \subseteq V$

---

1 : $v \in V_i$

2 : **While** $\mathcal{U}(v) > 0$ **do**

3 : **if** $\mathcal{U}(v)$ is largest among the vertices in V_i that are at a distance of at most 2 from v (ties are resolved by ID's) **then**

4 : v joins a restricted dominating set S_i

5 : **end if**

6 : **end while**

In the case of $V_i = V$, Algorithm I becomes the original distributed greedy algorithm given in [110] where it is shown that the algorithm returns a dominating set of size that is at most $(\ln \Lambda + 2)$ of the optimal in $O(n)$ time. Thus, using the similar approach as in [110], we get the following:

Proposition 1 *For a given $V_i \subseteq V$, if S^* is a minimum restricted dominating set with respect to V_i, then Algorithm I returns a restricted dominating set with respect*

to V_i of size at most $(\ln \Lambda + 2) \cdot |S^*|$ in $O(n)$ time. Here, Λ is the maximum degree of a graph.

Redundant Sensors:

For our original problem, a restricted dominating set with respect to V_i is computed by the actor network for each sensor type $i \in \mathbf{r}$. Thus, a subset of labels $s(v) \subseteq f(v)$ is determined for each $v \in V$, meaning that the vertices can get rid of some of the labels initially assigned to them while preserving the required condition $\bigcup_{u \in \mathcal{N}[v]} s(u) = \bigcup_{u \in \mathcal{N}[v]} f(u), \forall v \in V$. The assignment of label i to a vertex v represented that the corresponding actor v lies in the footprint of a sensor(s) of type i . Thus, if $i \notin s(v)$, then the sensor(s) of type i is redundant for the actor v, and deactivation of the sensor(s) will not affect the data collection by the actor v. This leads us to the next step in our scheme.

3.3.3 Deactivation of Redundant Sensors

S_i is the set of restricted dominating actors with respect to V_i as computed in Section 3.3.2. Thus, every $v \in S_i$ needs to have a sensor of type i transmitting data to v directly, i.e., v should be lying in the footprint of an active sensor of type i. In fact, actors in S_i are the only ones that need to receive data directly from a sensor of type i. Moreover, it is sufficient for $v \in S_i$ to receive data from only one such sensor. Thus, every $v \in S_i$ broadcasts a deactivating message to all the i-type sensors in v's footprint except for a single sensor (of type i) which receives an activating signal from v. The sensor receiving an activating message can be the one that is nearest to v. On the other hand, every $u \in (V - S_i)$ also broadcasts a deactivating signal to *all* the sensors of type i in u's footprint as u does not need to receive directly from an i type sensor. Sensors not receiving any of the activating or deactivating signal are the ones that do not lie within any actor's footprint, and are deactivated eventually. Also, an activating signal has a greater priority, thus, a sensor receiving an activating as well

as a deactivating message will become activated. This procedure will be performed for each type ($i \in \mathbf{r}$) of sensors.

After a fixed interval t_δ, all three steps (randomization, determination of redundant sensors, and deactivation of redundant sensors) are repeated. An example of the scheme is discussed in Section 3.5.

3.4 Analysis of the Random Distribution of Sensors

In WSANs, one way to characterize a random deployment of sensors of various types with λ_i intensities is to determine the number of actors that receive all types of data either directly from sensors, or by interacting with other actors. An exceeding percentage of such actors is highly desirable as it will imply an extended data access to the actors. In order to estimate this number, we proceed by introducing the following terms:

Definition 3.4.1 *In a colored graph G with $\mathbf{r} = \{1, 2, \cdots, r\}$ colors, A vertex v is said to be completely colored whenever*

$$\mathcal{H}(v) = \mathbf{r}$$

In other words, a vertex v is completely colored whenever it can find every color in the coloring set $\mathbf{r}$ in its closed neighborhood. Similarly, in terms of the actor network, we say that an actor v is *completely covered* whenever $\mathcal{H}(v) = \mathbf{r}$

We are interested in finding the probability of a vertex v being completely colored under the system model described in Section 3.1. It is to be recalled that the deployment of sensors of type i is modeled as a stationary spatial Poisson point process with constant intensity λ_i. The probability of having k sensors in an area A is then given by (e.g., [108]).

$$P_k = \frac{(\lambda_i A)^k e^{-\lambda_i A}}{k!} \tag{7}$$

Theorem 3.4.1 *For an actor v in the wireless sensor and actor network described in Section 3.1, the probability of the existence of an actor $u \in \mathcal{N}[v]$ such that u lies in the footprint of at least one sensor of type i for a given $i \in \mathbf{r}$ is*

$$P\left(i \in \mathcal{H}(v)\right) \;=\; 1 - e^{-\left[\lambda_i \alpha_i + \rho \alpha_a (1 - e^{-\lambda_i \alpha_i})\right]} \tag{8}$$

where λ_i and α_i are the intensity and the area of the footprint of sensor of type i respectively, whereas ρ and α_a are the intensity and the area of the footprint of actor respectively.

Proof.

$$P(i \in \mathcal{H}(v)) = 1 \;-\; P(i \notin \mathcal{H}(v))$$

$$P(i \notin \mathcal{H}(v)) = P(i \notin f(v)) \cdot \prod_{u \in \mathcal{N}(v)} P(i \notin f(u)) \tag{9}$$

Here, $P(i \notin f(v))$ is the probability that the label i is not assigned to the actor v. After inserting $k = 0$ and $A = \alpha_i$ in (7), we get $P(i \notin f(v)) = e^{-\lambda_i \alpha_i}$.

Similarly, $\prod_{u \in \mathcal{N}(v)} P(i \notin f(u))$ in (9) is the probability that none of the actors in the open neighborhood of v are assigned label i. We utilize (7) and standard results from stochastic geometry [108] to get

$$\begin{aligned}
\prod_{u \in \mathcal{N}(v)} P(i \notin f(u)) &= \sum_{n=0}^{\infty} P\left(\mid \mathcal{N}(v) \mid = n\right) \cdot \left[P\left(i \notin f(u)\right)\right]^n \\
&= \sum_{n=0}^{\infty} \frac{(\rho \alpha_a)^n \; e^{(-\rho \alpha_a)}}{n!} \; (e^{-\lambda_i \alpha_i})^n \\
&= (e^{-\rho \alpha_a})(e^{\rho \alpha_a e^{-\lambda_i \alpha_i}})
\end{aligned}$$

Inserting these values in (9) gives the following after some simplification.

$$P(i \notin \mathcal{H}(v)) \;=\; e^{-\left[\lambda_i a_i + \rho a' (1 - e^{-\lambda_i a_i})\right]}$$

Thus,

$$P\left(i \in \mathcal{H}(v)\right) \;=\; 1 - e^{-\left[\lambda_i \alpha_i + \rho \alpha_a (1 - e^{-\lambda_i \alpha_i})\right]}$$

■

Using the fact that the sensors of each type are deployed independent of each other, and therefore, colors are assigned to the vertices in a graph representing an actor network independent of each other, we deduce the following useful result.

Corollary 3.4.1 *The probability of an actor v to be completely covered in the wireless sensor and actor network described in Section 3.1 is*

$$\begin{aligned} P\left(\mathcal{H}(v) = \mathbf{r}\right) &= \prod_{i=1}^{|\mathbf{r}|} P\left(i \in \mathcal{H}(v)\right)) \\ &= \prod_{i=1}^{|\mathbf{r}|} \left[1 - e^{-\left[\lambda_i a_i + \rho a' (1 - e^{-\lambda_i a_i})\right]}\right] \end{aligned} \tag{10}$$

■

We observe that under the random distribution of sensors, $P\left(\mathcal{H}(v) = \mathbf{r}\right)$ can be improved by increasing λ_i and Δ_i for each $i \in \mathbf{r}$. However, increasing λ_i means increasing the number of sensors of type i, which is costly. Likewise, a higher Δ_i means sensors need to transmit farther requiring extra power. Thus, we aim to achieve a higher $P\left(\mathcal{H}(v) = \mathbf{r}\right)$ in an economical way (i.e., by keeping the number of active sensors low as well as smaller Δ_i). In fact, the energy-efficient data collection scheme described in Section 3.3 achieves this goal. The underlying objective is to determine all such sensors that are redundant in the sense that their deactivation will not affect the availability of data to the actors, and then eventually turn them off. Let X be the set of completely covered actors as a result of the random deployment of sensors of type i, $\forall i \in \mathbf{r}$. Using Proposition 1, we deduce that if S_i^* is the minimum number of sensors of type i that need to be activated to ensure that each actor in X is completely covered, then using simple distributed greedy algorithm (Algorithm I), our scheme makes every actor in X completely covered by activating at most $(\ln \Lambda + 2) \cdot |S_i^*|$ sensors of type i which is significantly smaller than the original number of deployed sensors of type i.

3.5 Example

Here, we present an example to illustrate the scheme discussed in Section 3.3. Consider a region with an area $\mathcal{A}$ in which sensors of three different types are distributed at random and independent of each other. Every sensor belongs to one of the types in $\mathbf{r} = \{1, 2, 3\}$. The distribution of sensors of each type $i \in \mathbf{r}$ is modeled as a Poisson point process with intensity λ_i. The radius of the footprint of a sensor of type i is Δ_i. For our example,

$$\lambda'_1 = 3 \quad \Delta_1 = 0.4$$
$$\lambda'_2 = 4 \quad \Delta_2 = 0.3$$
$$\lambda'_3 = 3 \quad \Delta_3 = 0.5$$

The actors are also distributed at random and independent of each other with intensity $\rho = 1.5$. Every actor has a footprint of radius $\Delta_a = 1$. An actor v interacts with all the actors lying within v's footprint. A graph representing interactions among actors along with the distribution of sensors is shown in Fig. 3.3.

In the *randomization* phase, a sensor of type i becomes activated with a probability p_i. Thus, the expected number of sensors of type i per unit area becomes $\lambda_i = p_i\lambda'_i$. Here, $p_1 = 0.6$, $p_2 = 0.4$, and $p_3 = 0.5$, therefore,

$$\lambda_1 = 1.8, \quad \lambda_2 = 1.6, \quad \lambda_3 = 1.5$$

The activated sensors after the randomization phase are shown in Fig. 3.3(b).

Next step is the *determination of redundant sensors* that can be deactivated by the actors without compromising the fact that if an actor v or one of its neighbors $u \in \mathcal{N}(v)$ lies within the footprint of a sensor of type i (i.e., v or some $u \in \mathcal{N}(v)$ is directly receiving data from a sensor of a type i) after the randomization phase, then v or some $u \in \mathcal{N}(v)$ must lie within the footprint of some sensor of type i even after the deactivation of redundant sensors.

As discussed in Section 3.3.2, redundant sensors can be determined by solving a restricted dominating set problem in the actor network. The process of determining

the redundant sensors of type 1 is shown in Fig. 3.4. The procedure is exactly same for the rest of the types of sensors. Let V_1 be the set of actors which lie within the footprint of at least one sensor of type 1 as shown in Fig. 3.4(a). The corresponding vertices in the graph modeling the actor network are assigned color (label) 1 as indicated by the colored vertices in Fig. 3.4(b). A restricted dominating set with respect to V_1 is then computed, as shown in Fig. 3.4(c). Finally, an actor corresponding to the vertex in the restricted dominating set with respect to V_1 activates a single sensor of type 1 within its footprint. All other sensors of this type are then *deactivated*, as shown in Fig. 3.4(d), leading to an energy-efficient data collection scheme.

The activated sensors of types $1, 2$, and 3 after the execution of all three phases, i.e., randomization, redundant sensors determination, and deactivation of redundant sensors, are shown in Fig. 3.5. We considered an area of $\mathcal{A} = 25$ (unit length)^2 for this example. The number of active sensors after different phases is given in Table 1.

Table 1: Number of active sensors after various phases.

	initially deployed	after randomization	after the deactivation of redundant sensors
Type 1	69	47	12
Type 2	104	35	10
Type 3	64	36	11

For a given Δ_i where $i \in \{1, 2, 3\}$, and $\lambda_1 = 1.8,\ \lambda_2 = 1.6$, and $\lambda_3 = 1.5$, the probability of an actor v being completely covered is 0.854 as computed from (10), i.e., $P\left(\mathcal{H}(v) = \{1, 2, 3\}\right) = 0.854$. There are 40 actors in total, out of which 36 (90%) are completely covered. Recall that if an actor v is completely covered actor, then for each $i \in \mathbf{r}$, there exists an actor $u \in \mathcal{N}[v]$ such that u lies within the footprint of at least one sensor of type i. Also note that for each sensor type in this example, more than two-third of the sensors that were active after the randomization phase are deactivated in the final step.

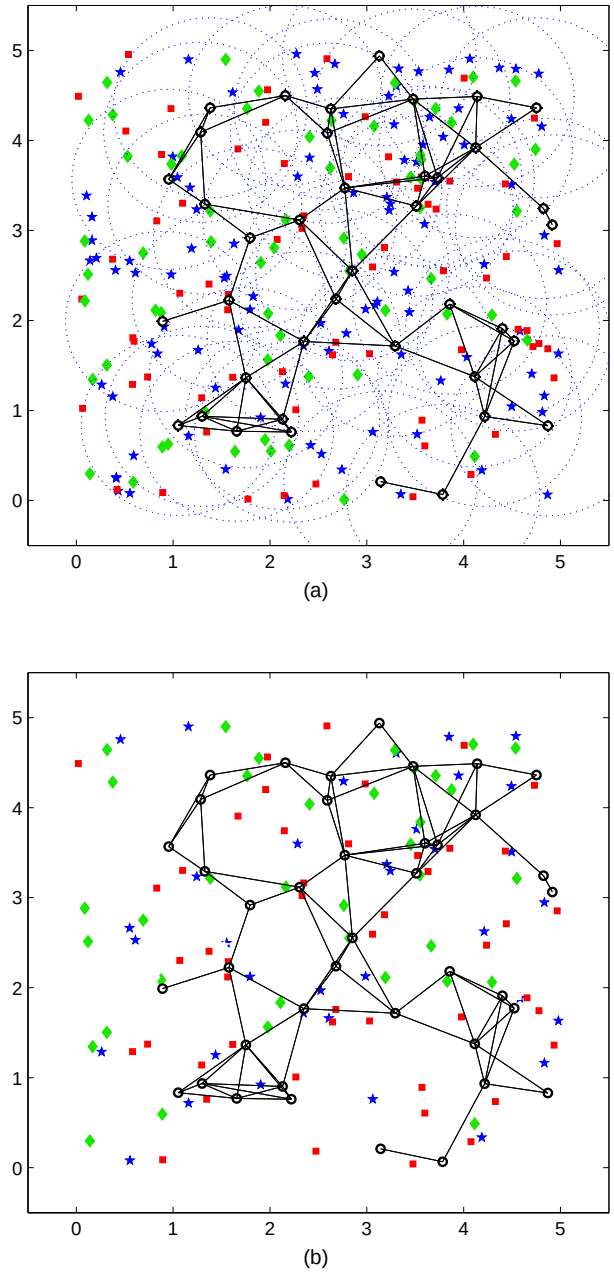


Figure 3.3: (a) Sensors of three types $1, 2$, and 3 (represented as $\square, \star$, and $\diamond$ respectively) are distributed at random along with an actor network represented by a graph. (b) Activated sensors after the randomization step are shown.

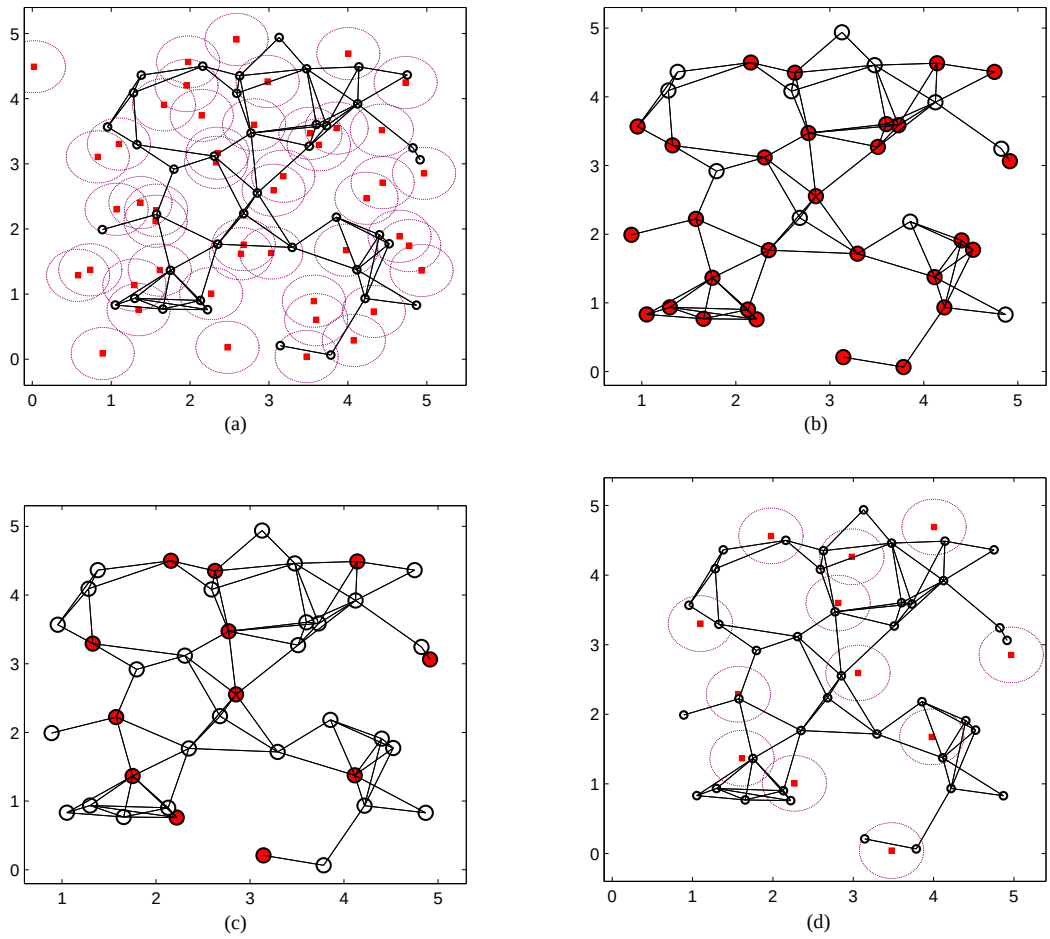


Figure 3.4: In (a) sensors of type 1 that are activated after the randomization are shown along with their foot prints. Actors lying within the footprints directly receive data from sensors, and are included in the set V_1. Vertices corresponding to these actors are colored in the graph of the actor network in (b). Thus, V_1 is the set of colored vertices in (b). In (c), a restricted dominating set of vertices with respect to V_1 is shown in the form of colored vertices. Each actor corresponding to a vertex in the restricted dominating with respect to V_1 activates a single sensor of type 1 within its footprint as shown in (d).

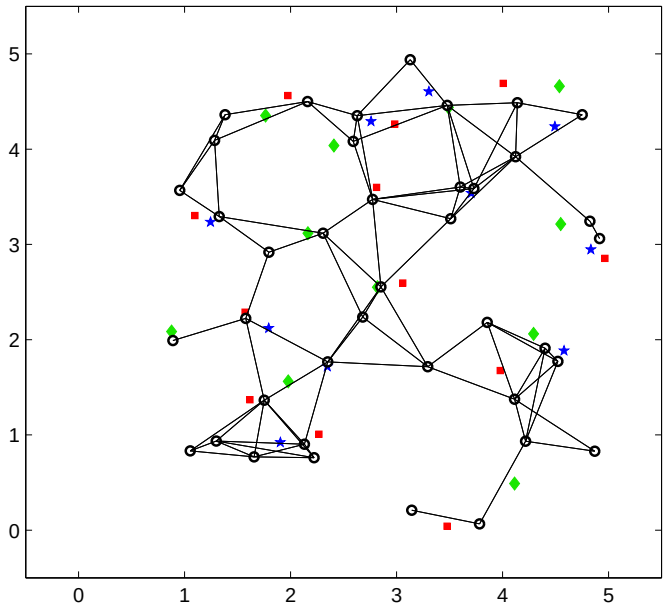


Figure 3.5: Sensors of each type 1,2, and 3 that are activated as a result of the energy-efficient scheme. All the redundant sensors (which were active initially) are deactivated here.

CHAPTER IV

HETEROGENEITY FOR EFFICIENT COMPLETE COVERAGE

In Chapters 2 and 3, we explored the promise heterogeneous networks hold for accomplishing complicated tasks by leveraging upon the assorted capabilities of agents. In this chapter, we exploit heterogeneity among agents for the complete coverage problem. The objective is to deploy sensors with circular sensing areas to completely cover a domain, i.e., every point in the domain should be covered by at least one sensor. Sensors can be arranged in many different ways to achieve complete coverage. However, we aim for configurations of sensors that provide an *efficient* solution to the problem. The most efficient configuration of sensors to completely cover a domain is known for the homogeneous case, i.e., when all sensors have exactly same sensing area. But, we show here that it is possible to further improve the efficiency of coverage by using sensors with different sensing areas.

Moreover, in the case of mobile agents, once a desired configuration is achieved, we aim to ensure that the configuration of agents, defined by the inter-agent distances, is not disturbed when agents move. In other words, agents move as a cohesive unit while maintaining the configuration and ensuring complete coverage. This demands distinctive coordination and information exchange among agents. We utilize constructs from graph theory and formation control to model this coordination. A comparison of formation costs of homogeneous and heterogeneous configurations is also performed. Thus, we have two major objectives here,

(1) construct and analyze configurations of agents with circular sensing areas of two different radii that can provide complete coverage with improved coverage

efficiencies as compared to the case when sensing areas of all agents have same radii.

(2) determine coordination framework among agents so that they retain their formation in a particular configuration while they exhibit movements.

This chapter is organized as follows: a formal description of the efficiency of coverage in terms of the coverage density and sensing cost is given in Section 4.1. In Section 4.2, configurations of circular disks with same and different radii are illustrated. Coverage densities and sensing costs are computed for heterogeneous configurations alongside a comparison with the homogeneous case in Section 4.3. In Sections 4.4 and 4.5, formation cost is introduced and formation graphs corresponding to different configurations are obtained.

4.1 Efficiency of Coverage

It is already mentioned that our objective is to completely cover a region, i.e., there should be no holes present. Sensors with given sensing footprints can be arranged in many different ways to achieve this objective. However, the efficiency of the coverage will vary for different configurations in terms of the power consumed to completely cover a given domain. A *circle covering* is a configuration of overlapping circles with given radii to completely cover some domain in $\mathbb{R}^d$ ($d = 2$ in our case).

In order to quantify the efficiency of circle covering, a commonly used metric is the *covering density*, denoted by ρ. It measures how efficiently circular disks are arranged to cover the given domain. The notion of *crystallographic unit* [115] of the circle covering is needed to define ρ. A *crystallographic unit* of the covering is a parallelogram containing the minimum repeatable elements of a circle covering.

Definition 4.1.1 *(Covering Density) The covering density of a circle covering, denoted by ρ, is the ratio of the area of the crystallographic unit to the area of circular disks covering the crystallographic unit.*

Let c be the crystallographic unit of a circle covering and $A(c)$ be the area of c,

then

$$\rho = \frac{A(c)}{\sum_{i=1}^{k} x_i \left(\pi \delta_i^2\right)} \tag{11}$$

k is the number of circular disks that are covering c. x_i is the fraction of the area of the circular disk of radius δ_i covering a certain portion of c. An example is shown in Fig. 4.1.

A circle covering with a higher ρ is more efficient. It means that the overlap among the circular disks is smaller and the areas of disks are being utilized efficiently to cover the region.

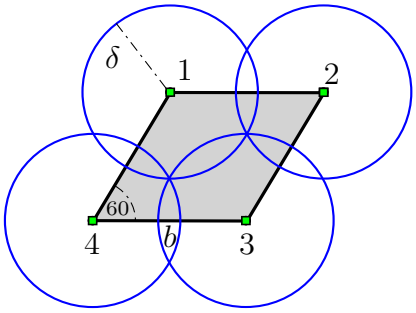


Figure 4.1: Four disks of equal radii are taking part in covering a crystallographic unit of area $\frac{\sqrt{3}}{2}b^2$. The areas of circular discs $1, 2, 3$, and 4 covering a portion of crystallographic unit are $\frac{\pi\delta^2}{3}$, $\frac{\pi\delta^2}{6}$, $\frac{\pi\delta^2}{3}$, and $\frac{\pi\delta^2}{6}$ respectively. $\rho = \frac{\sqrt{3}b^2}{2\pi\delta^2}$

In practical scenarios, these circular disks may correspond to sensing footprints of the sensors deployed within some region. Overall energy consumed by sensors for sensing operations is directly dependent on the area of footprints. In fact, in a sensor range model based on RF-power density function for an isotropic antenna, power consumed is directly proportional to the area of the sensor footprint given by $\pi(\delta_i)^2$ where δ_i is the radius of the footprint (e.g., [112, 113]). Thus, a sensing cost (in terms of the power consumption) can be associated with a sensor that is proportional to the square of the radius of the footprint of the sensor. To efficiently utilize the energy resources, sensors should be deployed to maximally utilize the area of circular footprints. Therefore, we associate a sensing cost with a configuration of a group of

agents covering a region through a circle covering.

Definition 4.1.2 *(Sensing Cost) Let there be n agents with each agent having a sensing footprint of a circular shape with a radius δ_i, $i \in \{1, 2, \cdots, n\}$. We define the overall sensing cost of the system as,*

$$\mathcal{J} = \sum_{i=1}^{n} (\delta_i)^2 \tag{12}$$

Like the density, $\mathcal{J}$ is also a measure of the efficiency of the circle covering from the energy consumption perspective. If τ_1 and τ_2 are two configurations with exactly same number of nodes and covering the same region such that $\mathcal{J}_{\tau_1} < \mathcal{J}_{\tau_2}$, then τ_1 is indeed a better choice from the sensing cost view point. A large amount of overlap among circles in a circle covering results into an increase in the sensing cost.

Covering density and Sensing cost, both will be used to compute the efficiency of circle coverings presented in the subsequent sections.

4.2 Agent Configurations for Circle Covering

Circle coverings can be achieved through circular disks all having same radii (homogeneous configurations), or through circular disks with different radii (heterogeneous configurations). In this section, we present details of homogeneous configuration and propose two heterogeneous configurations of agents to completely cover a region.

4.2.1 Homogeneous Configuration

Let there be a group of agents each having a sensing footprint of radius δ. We want to arrange them within a region to obtain a circle covering of the region with the minimum density. It is shown in [67] that the optimal way to achieve this is to place agents on an equilateral triangle lattice as shown in Fig. 4.2. The lines connecting agents whose sensing footprints intersect form an equilateral triangle with the length of each side being equal to $\sqrt{3}\delta$. Moreover, the density of the covering, denoted by

ρ_h, is $\frac{\sqrt{27}}{2\pi}$ which is the best possible[1]. We call this configuration as the *homogeneous configuration* as every agent has a sensing footprint of exactly same radius δ. If there are n agents, the sensing cost of the homogeneous configuration will be $\mathcal{J}_h = n\delta^2$.

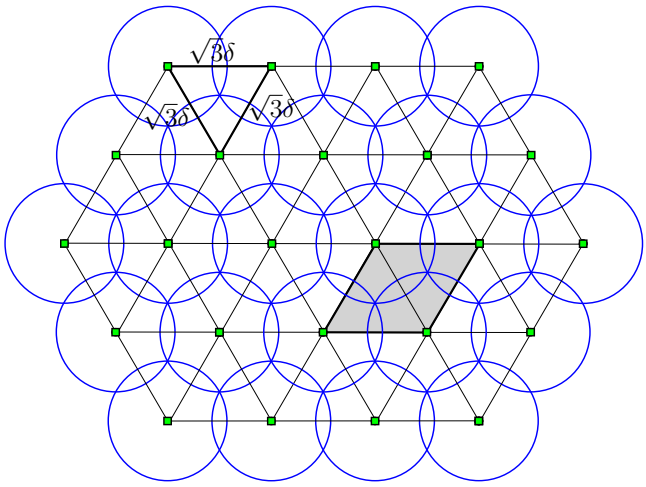


Figure 4.2: Circle covering through circular disks of same radii. The shaded region is a crystallographic unit.

4.2.2 Heterogeneous Configurations

Using heterogeneous disks (in terms of the radii) for the purpose of circle covering may further improve the efficiency of covering. Thus, we present two heterogeneous configurations here. It will be shown in the next section that they are indeed better than the homogeneous case from the covering density and the sensing cost perspectives.

4.2.2.1 Configuration 1 (Γ_1)

Let there be two types of agents with respect to the radius of the sensing footprint. The first type has footprints of radius δ_1, whereas second type of agents have $\delta_2 < \delta_1$ as the radius of their sensing footprints. These two types of agents are placed on a grid network alternatively as shown in Fig. 4.3. If the length of the side of each square in the grid is a, then Γ_1 configuration results into a circle covering of the region

[1]It is assumed that the area being monitored is much larger than the sensing print of an individual agent.

if δ_1 and δ_2 satisfies the following:

$$\begin{aligned} \delta_1 &= \epsilon_1 a, & \tfrac{1}{\sqrt{2}} \leq \epsilon_1 \leq 1 \\ \delta_2 &= \tfrac{1}{\sqrt{2}}\left(a - \sqrt{2\delta_1^2 - a^2}\right) \end{aligned}$$

A crystallographic unit of the circle covering as a result of Γ_1 is shown in Fig. 4.3. Thus, the density of the covering, denoted by ρ_1 is

$$\rho_1 = \frac{\text{(crystallographic unit area)}}{\pi(\delta_1^2 + \delta_2^2)} = \frac{2a^2}{\pi(\delta_1^2 + \delta_2^2)} \tag{13}$$

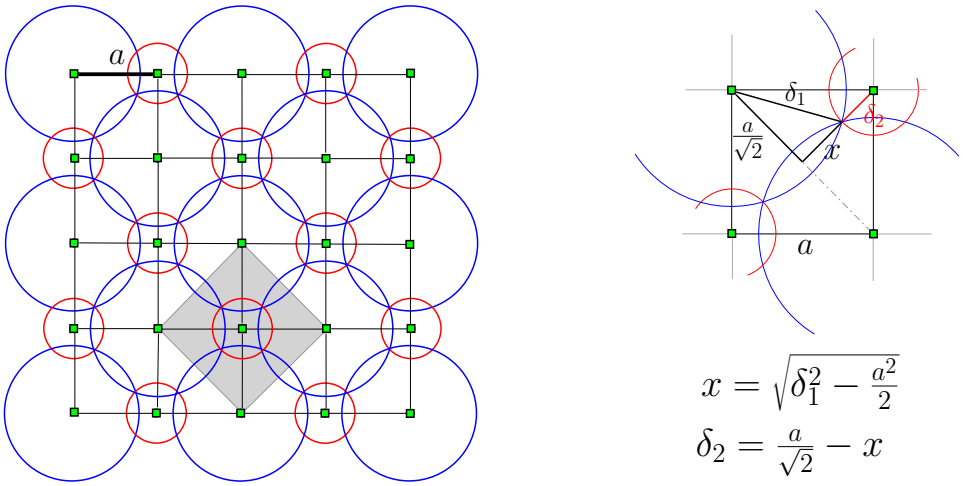


Figure 4.3: Heterogeneous configuration 1 in which the shaded parallelogram is a crystallographic unit of the covering. For a given δ_1, the appropriate value of δ_2 is also computed.

4.2.2.2 Configuration 2 (Γ_2)

As in the case of Γ_1, we consider two types of agents; one with sensing footprints of radius δ_1, and the other having radius δ_2, where $\delta_2 < \delta_1$. These agents are placed on an equilateral triangle lattice as shown in Fig. 4.4. Let b be the length of the side of the unit equilateral triangle in the network, then δ_1 and δ_2 need to satisfy the following conditions in order to obtain a circle covering from this configuration of agents as illustrated in Fig. 4.4.

$$\begin{aligned} \delta_1 &= \epsilon_2 b, & \frac{\sqrt{3}}{2} \leq \epsilon_2 \leq 1 \\ \delta_2 &= \frac{1}{2}\left[b - \sqrt{4\delta_1^2 - 3b^2}\right] \end{aligned}$$

The density of Γ_2, denoted by ρ_2, can be obtained by computing the area of the crystallographic unit of the covering (as shown in Fig. 4.4). It should be noted that the area of circles covering a crystallographic unit is equal to the sum of the area of a circle with δ_1 radius and areas of two circles with δ_2 radius.

$$\rho_2 = \frac{\text{(crystallographic unit area)}}{\pi(\delta_1^2 + 2\delta_2^2)} = \frac{\frac{3\sqrt{3}}{2}b^2}{\pi(\delta_1^2 + 2\delta_2^2)} \tag{14}$$

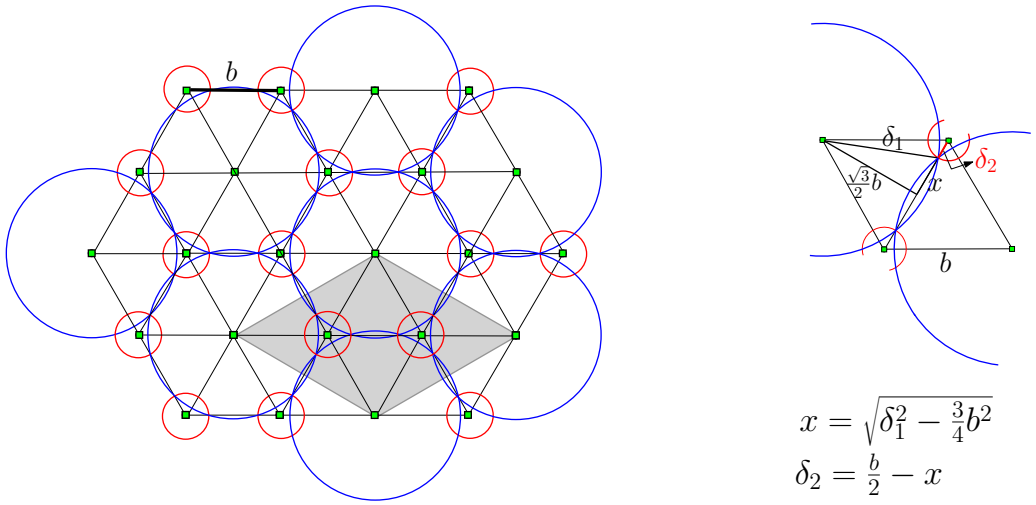


Figure 4.4: Heterogeneous Configuration 2. A crystallographic unit is a shaded parallelogram. A relationship between δ_1 and δ_2 to obtain a circle covering is also shown.

4.3 Comparison of Homogeneous and Heterogeneous Configurations

In this section, we compute and compare coverage densities and sensing costs of configurations introduced in Section 4.2, thereby establishing that heterogeneous configurations can outperform homogeneous configuration.

4.3.1 Comparison of Coverage Densities

The maximum coverage density for homogeneous case is $\rho_h = \frac{\sqrt{27}}{2\pi}$. The covering densities for heterogeneous configurations 1 and 2 are given in (13) and (14) respectively. ρ_1 and ρ_2 both depend on δ_1 and δ_2. Also, δ_2 can be computed once an appropriate δ_1 is selected. $\delta_1 = \epsilon_1 a$ for the heterogeneous configuration 1, whereas $\delta_1 = \epsilon_2 b$ for the heterogeneous configuration 2. Valid ranges for ϵ_1 and ϵ_2 are given in Section 4.2.2. Thus, if we define radii ratio of disks as $\alpha = \delta_2/\delta_1$, then the covering density directly depends on α. Note that $\alpha = 1$ for the homogeneous case.

If we plot covering densities of homogeneous as well as heterogeneous configurations with respect to α, we get the plots shown in Fig. 4.5. It can be seen that for quite a good range of α, heterogeneous configurations have better covering densities than the homogeneous case specially when α is smaller. The maximum covering density obtained with the heterogeneous configuration 1 is 0.849 with $\alpha = 0.447$. Similarly, maximum density for heterogeneous configuration 2 is obtained when $\alpha = 0.18$ and it is 0.902, which is significantly higher than the homogeneous case.

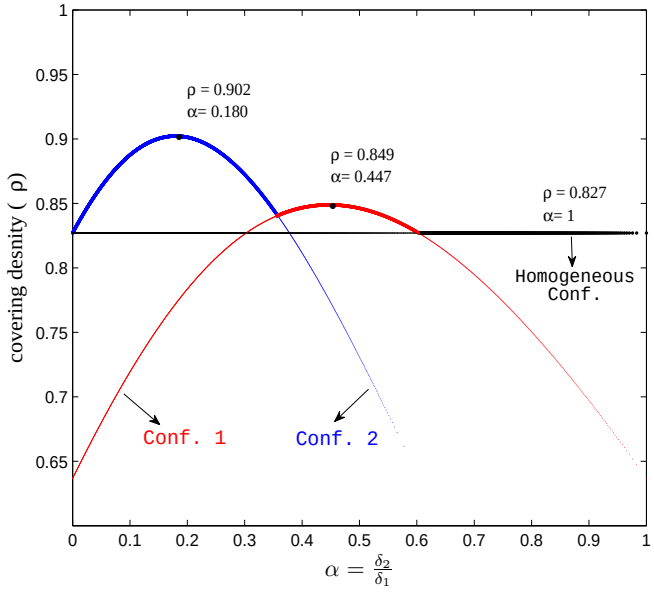


Figure 4.5: Plots of covering densities of different configurations as a function of α.

4.3.2 Comparison of Sensing Costs

Now, we compare homogeneous configuration with heterogeneous configurations using sensing cost metric defined in (12).

In homogeneous configuration, nodes are placed in an equilateral triangle network. Let τ be an $m \times n$ equilateral triangle network[2] in which b is the distance between two adjacent nodes. The area covered by the network is given by

$$A_\tau \;=\; 2(n-1)(m-1)\frac{\sqrt{3}}{4}b^2 \tag{15}$$

In heterogeneous configuration 1, nodes are placed in a grid network. Consider an $m \times n$ square grid network in which the distance between any two adjacent nodes is a, then the area of the grid is

$$A_g \;=\; (n-1)(m-1)a^2 \tag{16}$$

Since we want to compare the sensing cost of homogeneous configuration with that of the heterogeneous configuration 1, we need a grid network with the same number of nodes and area as the triangular network τ. Thus, for an equilateral triangle network with a given b, we obtain an equivalent grid network with a by equating (15) and (16).

$$a \;=\; \left(\frac{3}{4}\right)^{\frac{1}{4}} b \tag{17}$$

Now, we compute the sensing cost of homogeneous configuration of nodes, denoted by $\mathcal{J}_h$, in which nodes with sensing footprints of radii $\delta = \frac{b}{\sqrt{3}}$ are placed on the vertices of an $m \times n$ triangle network τ to obtain a circle covering of an area A_τ.

[2]nodes are arranged in an equilateral triangle network in such a way that there are m rows of nodes with n nodes in each row.

$$\mathcal{J}_h \;=\; \sum_{i=1}^{mn} \left(\frac{b}{\sqrt{3}}\right)^2 \;=\; \frac{mnb^2}{3} \tag{18}$$

Next, nodes with sensing footprints of radii $\delta_1 = \epsilon_1 a$, and $\delta_2 = \frac{1}{\sqrt{2}}\left(a - \sqrt{2\delta_1^2 - a^2}\right)$, where $\epsilon_1 \in \left[\frac{1}{\sqrt{2}} \;\; 1\right]$ and a is given by (17), are placed in the heterogeneous configuration 1 to obtain a circle covering of an area A_g $(= A_\tau)$. The sensing cost of such a configuration is [3]

$$\begin{aligned}
\mathcal{J}_1(\epsilon_1) =& \frac{mn}{2}\left[(\delta_1)^2 \;+\; (\delta_2)^2\right] \\
=& \frac{mn}{2}a^2 \;\left[2\epsilon_1^2 \;-\; \sqrt{2\epsilon_1^2 \;-\; 1}\right] \\
=& \frac{\sqrt{3}}{4}mnb^2 \;\left[2\epsilon_1^2 \;-\; \sqrt{2\epsilon_1^2 \;-\; 1}\right]
\end{aligned} \tag{19}$$

In fact, $\mathcal{J}_1(\epsilon_1) < \mathcal{J}_h$ when $0.751 \leq \epsilon_1 \leq 0.84$. The minimum $\mathcal{J}_1$ is obtained for $\epsilon_1^* = 0.7906$. Moreover, for $\delta_1 = \epsilon_1^* a$, we get $\alpha = 0.447$ which is the radii ratio that gives the maximum covering density for the heterogeneous configuration 1. Computing the ratio of $\mathcal{J}_1(\epsilon_1^*)$ to $\mathcal{J}_h$ gives,

$$\frac{\mathcal{J}_1(\epsilon_1^*)}{\mathcal{J}_h} = \frac{9\sqrt{3}}{16} \;<\; 1 \tag{20}$$

Thus, heterogeneous configuration 1 can be more economical than the homogeneous configuration from a sensing cost perspective.

Similarly, a comparison between homogeneous and heterogeneous configuration 2 can be performed by placing nodes with sensing footprints of radii $\delta_1 = \epsilon_2 b$ and $\delta_2 = \frac{1}{2}\left[b - \sqrt{4\delta_1^2 - 3b^2}\right]$ on the vertices of an $m \times n$ equilateral triangle network to get a circle covering. Here, $\epsilon_2 \in \left[\frac{\sqrt{3}}{2} \;\; 1\right]$. The sensing cost of this configuration of nodes is

[3] In heterogeneous configuration 2, there are twice as many circles with δ_2 radius as compared to circles with δ_1 radius as long as both $\pi(\delta_1)^2$ and $\pi(\delta_2)^2$ are significantly smaller than A_g.

$$\begin{aligned}\mathcal{J}_2(\epsilon_2) =& mn\left[\left(\frac{1}{3}\right)\delta_1^2 \ + \ \left(\frac{2}{3}\right)\delta_2^2\right] \\ =& \frac{mnb^2}{3}\left[3\epsilon_2^2 \ - \ 1 \ - \ \sqrt{4\epsilon_2^2 - 3}\right]\end{aligned} \tag{21}$$

For $0.882 \leq \epsilon_2 < 1$, $\mathcal{J}_2(\epsilon_2) < \mathcal{J}_h$. In fact, $\mathcal{J}_2$ is minimum at $\epsilon_2^* = 0.928$. Maximum covering density for heterogeneous configuration 2 is also achieved when $\delta_1 = \epsilon_2^* b$. Computing the ratio of $\mathcal{J}_2(\epsilon_2^*)$ to $\mathcal{J}_h$ gives,

$$\frac{\mathcal{J}_2(\epsilon_2^*)}{\mathcal{J}_h} = \frac{11}{12} \ < \ 1 \tag{22}$$

Thus, heterogeneous configuration 2 is more efficient than the homogeneous one if an appropriate value of ϵ_2 is chosen.

Proposition 2 *It is possible to have a lower sensing cost and a higher covering efficiency for the circle coverings obtained through heterogeneous configurations* 1 *and* 2 *as compared to the homogeneous configuration. In fact,*

$$\begin{aligned}&\mathcal{J}_1(\epsilon_1) < \mathcal{J}_h, \quad \rho_1 > \rho_h, \quad \textit{for} \quad 0.751 \leq \epsilon_1 \leq 0.84 \\ &\mathcal{J}_2(\epsilon_2) < \mathcal{J}_h, \quad \rho_2 > \rho_h, \quad \textit{for} \quad 0.882 \leq \epsilon_2 \leq 1\end{aligned}$$

Moreover,

$$\mathcal{J}_2(\epsilon_2^*) < \mathcal{J}_1(\epsilon_1^*) < \mathcal{J}_h \tag{23}$$

where $\epsilon_1^* = 0.7906$ *and* $\epsilon_2^* = 0.928$.

■

4.4 Persistent Formations of Agents

In this section, we address the second part of our main problem, i.e., to obtain coordination frameworks that will allow nodes to maintain a desired configuration (homogeneous or heterogeneous ones) even when nodes execute movements, and analyze

the cost associated with those frameworks. This issue of maintaining specific node configurations under motions is directly related to the notion of formation control. In order to maintain the overall formation of a network of agents, agents implement local control laws taking into account the relative position of their neighbors while ensuring that the inter-agent distances are held constant while they are moving.

Graph-theoretic tools have been effectively employed to obtain formation control architectures in which maintaining a subset of inter-agent distances ensure that the overall formation is maintained. The notion of graph rigidity has been a key object in this regard. If a shape of a formation of a group of n agents is defined by inter-agent distances, $\mathcal{D} = \{d_{ij} \in \mathbb{R}_+ \mid \ i, j = 1, 2, \cdots, n, \ i \neq j\}$, then roughly speaking a *rigid graph* $G(V, E)$ will ensure that maintaining inter-agent distances only among the adjacent nodes in G will preserve all the distance constraints in $\mathcal{D}$. For precise definitions and results regarding algebraic and combinatorial characterizations of rigidity, see [116, 118] and the references therein for details.

In order to maintain distance d_{ij}, nodes i and j implement formation control laws that require them to sense the relative positions of each other. In an undirected case (rigid graph), the task of maintaining d_{ij} is executed by both nodes i and j. However, it is possible to assign this task to only one of the nodes [117], thus making an edge between i and j a directed one. In this situation, only one of the two nodes needs to sense the relative position of the other node. Therefore, for the formation control purposes, it is possible to obtain a directed graph in which a directed edge from node i to j would mean that i needs to sense the relative position of j and is responsible to satisfy the distance constraint d_{ij}. Such a directed graph is known as the *persistent graph*. It is to be mentioned that assigning appropriate directions to undirected edges in a rigid graph to obtain a directed persistent graph is not a straight-forward task and requires an in-depth understanding of the problem (e.g., see [116, 117, 118] for details).

An agent measures relative positions of a subset of other agents (defined by the persistent graph) for the formation control purpose. Power is consumed to accomplish this task. More power is required if neighboring agents are farther away from the agent. In fact, if agent i has a distance constraint of Δ_i where Δ_i is the longest distance between agent i and its farthest out-going neighbor, then power consumed by i is directly proportional to $(\Delta_i)^2$ [112]. Thus, we define formation cost as follows:

Definition 4.4.1 *(Formation Cost) Let G be a directed graph with n vertices and Δ_i be the length of the longest out-going edge from vertex v_i, then the formation cost is defined as*

$$\mathcal{F} = \sum_{i=1}^{n} (\Delta_i)^2 \tag{24}$$

Now, we find persistent formation graphs with the minimum formation costs for the homogeneous configuration as well as the heterogeneous configurations and then compare their formation costs.

4.4.1 Comparison of Formation Costs

Since homogeneous and heterogeneous configuration 2, both correspond to the arrangement of nodes in an equilateral triangle lattice, persistent formation graph for both configurations will be same. Thus, formation cost will also be same in both cases. In an undirected graph corresponding to an equilateral triangle lattice (as shown in Figs. 4.2 and 4.4), every edge is of same length, say b. It is known that such a graph is rigid (e.g., [120]). Thus, there exists a minimally rigid[4] subgraph of an equilateral triangle graph in which every edge is of the same length. It is shown in [118] that it is always possible to assign directions to edges in a minimally rigid graph to obtain a directed minimally persistent graph. Thus, we get a minimally persistent graph for an equilateral triangle formation in which the length of the longest out-going edge

[4]A rigid graph is called as *minimaly rigid* if the removal of any edge makes it non-rigid.

from any node will be same, i.e., $\Delta_i = b,\ \forall\ i$. Thus, for an $n \times n$ equilateral triangle lattice, formation costs for homogeneous and heterogeneous configuration 2 is,

$$\mathcal{F}_h \ = \ \mathcal{F}_2 \ = \ (nb)^2 \tag{25}$$

In order to find the formation cost of the heterogeneous configuration 1, first we need to find a persistent graph with the minimum formation cost for a grid formation. The construction of such a graph for an $n \times n$ grid in which the length of the side of each square in the grid is a is given in Section 4.5. It is shown that there will be $(n-1)$ nodes with $\sqrt{2}a$ as the length of the longest out-going edge from them, whereas rest of the $(n^2 - (n-1))$ nodes will have a as Δ_i. Thus, the formation cost for such a graph will be,

$$\begin{aligned} \mathcal{F}_1 \ &= (n-1)(\sqrt{2}a)^2 \ + \ (n^2 - n + 1)a^2 \\ &= a^2(n^2 + n - 1) \end{aligned} \tag{26}$$

In order to compare $\mathcal{F}_1$ with $\mathcal{F}_h$, we replace a in (26) with b in (17) to get

$$\mathcal{F}_1 \ = \ \frac{\sqrt{3}}{2} b^2 (n^2 + n - 1) \tag{27}$$

Using (25) and (27), we get the following:

Proposition 3 *Let $\mathcal{F}_h$, $\mathcal{F}_1$, and $\mathcal{F}_2$ be the formation costs of persistent formation graphs corresponding to homogeneous and heterogeneous configurations* 1*, and* 2 *respectively. In the case of n^2 nodes arranged in the form of an $n \times n$ grid (equilateral triangular grid or square grid),*

$$\begin{gathered} \mathcal{F}_2 \ = \ \mathcal{F}_h, \qquad \textit{and} \\ \mathcal{F}_1 \ < \ \mathcal{F}_h \quad \textit{whenever} \quad (n-1) < \left(\tfrac{2-\sqrt{3}}{\sqrt{3}} \right) n^2 \end{gathered}$$

■

It is to be mentioned here that for an $m \times n$ grid, where $m \neq n$, $\mathcal{F}_2 = \mathcal{F}_h$ still holds, and the condition to have $\mathcal{F}_1 < \mathcal{F}_h$ can be obtained using a similar approach.

4.5 Minimally Persistent Graph for a Grid Formation

A grid graph shown in Fig. 4.6(a) is not rigid. It can be made rigid by adding diagonal edges, sometimes called as *braces*, to the squares in the grid as shown in Fig. 4.6(b). In order to determine the braces that will make a grid minimally rigid, a *brace graph* is defined (e.g., see [119]). For an $n \times n$ grid, a brace graph is a bi-partite graph with $(n-1)$ vertices in each of the two partitions of the bi-partite graph. Vertices in one partition, denoted by $\{r_1, \cdots, r_{(n-1)}\}$ correspond to the rows of the grid, while vertices corresponding to the columns of the grid are denoted by $\{c_1, \cdots, c_{(n-1}\}$ and are included in the second partition of the brace graph. In the brace graph, r_i is connected to c_j whenever a brace (diagonal edge) exists in the i^{th} row and the j^{th} column of the grid. An example is shown in Fig. 4.6(c). An important result regarding the minimal rigidity of grid graphs is

Theorem 4.5.1 *[119] A grid is minimally rigid whenever the corresponding brace graph is a tree.*

The above theorem implies that $(2n-3)$ braces need to be added in an $n \times n$ grid to make it minimally rigid. The brace graph solves the minimal rigidity problem for undirected grid graphs, but we aim to construct a directed grid graph that is persistent with the minimal formation cost. As mentioned earlier, it is always possible to assign directions to the edges in a minimally rigid graph to obtain a directed minimally persistent graph. Moreover, the following result will be used to obtain such an assignment of direction to the edges.

Theorem 4.5.2 *[117] A graph is minimally persistent whenever it is minimally rigid and every vertex has an out-degree*[5] *of at most two.*

[5] *Out-degree of a vertex v, denoted by $d^+(v)$ is the number of out-going edges from v.*

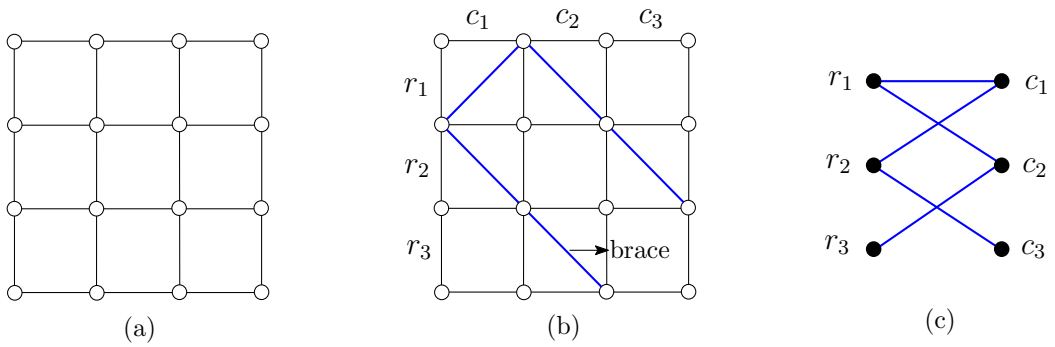


Figure 4.6: (a) A non-rigid grid graph. (b) Braces are added to the grid graph. (c) The corresponding brace graph.

Thus, we need to assign directions to the edges in an undirected grid graph with braces such that the out-degree of any vertex is at most 2, i.e., $d^+(v) \leq 2$. Moreover, there will always be $2(n-1)$ vertices that have a brace as an out-going and/or in-coming edge. Since every brace is an edge with the longest length in the grid graph with braces, we want to minimize the number of nodes that have an out-going brace to minimize the formation cost in (24). Now, under any assignment of directions to edges, there will always be at least $(n-1)$ nodes that have an out-going brace. Thus, our objective is to obtain a directed graph from an undirected grid graph with braces such that

(i) $d^+(v) \leq 2$, and

(ii) the number of vertices to have braces as out-going edges is $(n-1)$.

We provide such a construction of a directed grid graph below.

4.5.1 Construction of Minimally Persistent Grid Graphs

(1) From an $n \times n$ undirected grid graph, get a directed graph by assigning all the horizontal edges a direction from left to right and all vertical edges a direction from upwards to downwards. Note that all of the vertices have an out-degree of 2 except $(2n-1)$ boundary nodes which are denoted by $k_j,\ j \in \{1, 2, \cdots, (2n-1)\}$ as shown in Fig. 4.7(a).

(2) Add braces such that the braces and their end vertices, denoted by $b_1, b_2, \cdots, b_{2(n-1)}$,

constitute a path graph as shown in Fig. 4.7(b). (The corresponding brace graph is also a path graph.)

(3) Assign directions to the braces such that braces from the vertices b_{2i} $i \in \{1, 2, \cdots, n-1\}$ are directed outwards. An example is shown in Fig. 4.7(c). Note that $d^+(v) = 4,\ \forall v \in \{b_2, b_4, \cdots, b_{2(n-1)}\}$. But, we want every vertex to have a maximum out-degree of 2.

(4) For every $b_{2i},\ i \in \{1, 2, \cdots, (n-1)\}$, find two disjoint directed paths (which is always possible) from b_{2i} to two distinct k_j's, $j \in \{1, 2, \cdots, 2n-1\}$, where k_j is a node with an out-degree strictly less than 2. If $b_{2i} = k_j$ for some $j \in \{1, 2, \cdots, (2n-1)\}$, then only one directed path is needed from $b_2 i$.

(5) Reverse the directions of all the edges included in the paths obtained in step (4).

An illustration of the above scheme is given in Fig. 4.7. The scheme gives a directed graph in which $d^+(v) \leq 2,\ \forall\ v$ and only $(n-1)$ vertices have out-going braces which is the best possible in terms of minimizing the number of vertices that have braces as out-going edges. Thus, we get the following:

Proposition 4 *The scheme presented in Section 4.5.1 gives a minimally persistent graph with the minimum formation cost for the formation of nodes in an $n \times n$ square grid.*

In this chapter, we examined the complete coverage problem for mobile agents with circular sensing areas. It is shown that more efficient solutions can be obtained through networks of heterogeneous agents as compared to the homogeneous one. In particular, configurations of agents with two different radii are examined that exhibit higher coverage densities and lower sensing costs than the configuration of agents having same radii. In the case of heterogeneous configuration 1, best results are obtained when sensing areas of one type of agents are approximately half (0.447)

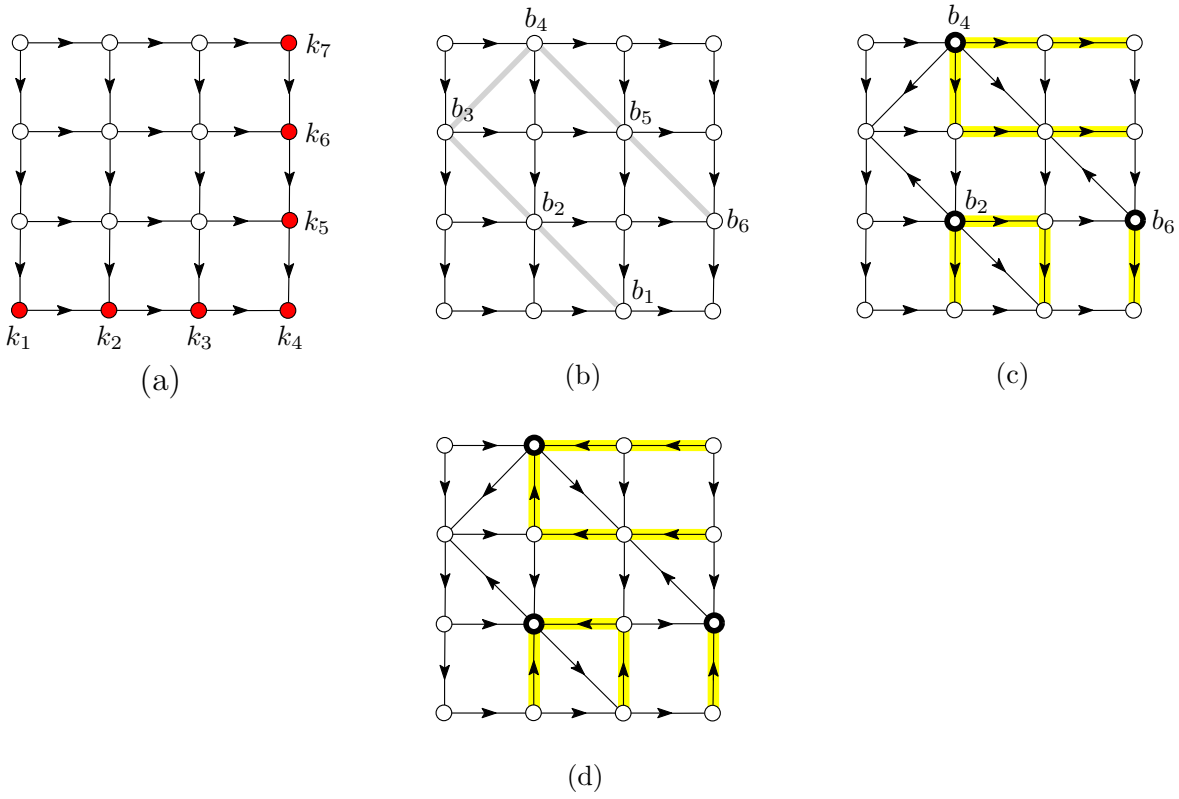


Figure 4.7: (a) Step 1 - assignment of directions to the edges of grid graph. (b) Step 2 - addition of braces. (c) Steps 3 & 4 - assignment of directions to the braces and obtaining disjoint directed paths (d) Step 5 - reversal of the directions of the edges in the directed paths obtained in step 4.

of the size of others, whereas in heterogeneous configuration 2, optimum coverage density and sensing cost are obtained when the size of one disk is approximately one-fifth (0.18) of the size of the other. Moreover, coordination frameworks, which are necessary to ensure the formation of agents in desired configurations while agents move, are also analyzed.

CHAPTER V

SECURING A NETWORK AGAINST A SEQUENCE OF INTRUDER ATTACKS

So far, we have focused on exploiting underlying network topologies to allow agents maximally utilize resources available within networks through local interactions. The issue of making networks reliable and resilient is another integral matter of the network design process, and is closely related to the design of underlying graph structures of networks. In Chapters 5 and 6, we investigate role of network topologies for reliable operations of networks in terms of securing networks from external intrusions and characterizing robustness properties of networks.

We start with the topic of securing a network of agents against a sequence of intruder attacks through heterogeneous guards in this chapter. A network of agents is represented by a graph G, in which the vertex set V represents agents, and the edge set E corresponds to interactions among agents. A set of guards, where each guard is capable of detecting and responding to an intruder attack within a certain distance, is distributed across the network. We require the graph to be secured in the sense that each intruder attack can be responded to sufficiently fast by some guard, and all the vertices in the graph remain secure even after the movement of a guard to an attacked vertex. All the components of this problem, including the number of guards required, the deployment of guards among various nodes, and the movement strategies for guards to counter intruder attacks, are addressed here.

We begin by introducing various terms that will be used throughout this section. An edge between vertices u and v is denoted by $u \sim v$. A set $I \subset V(G)$ is an *independent set* of a graph if no two vertices in I are adjacent in G. The *independence*

number, $\iota(G)$, is the cardinality of a largest independent set. A *complete* subgraph is induced by the vertices in a set $W \subseteq V$ whenever $u \sim v \in E$ for all $u, v \in W$. A subset of vertices inducing a complete subgraph is referred to as a *clique*. A clique that can not be extended by including one or more adjacent vertices is a *maximal clique*. In other words, a maximal clique is a clique that is not contained in a larger clique in the graph. Finding all maximal cliques in a graph is known as a *maximal clique decomposition* of a graph. The distance between two vertices u and v in G, denoted by $d(u, v)_G$, is the length of the shortest path between u and v in G. *Path length* is the number of edges in a path. A path length of r is sometimes referred to as an *r-hop*. The *diameter* of a graph, $diam(G)$, is $\max\ d(u, v)_G,\ \forall u, v \in G$. The r^{th} *power of a graph* G, denoted by G^r, is a graph with $V(G^r) = V(G)$, and $u \sim v \in E(G^r)$, whenever $d(u, v)_G \leq r$.

5.1 Problem formulation

In this section, we formulate the problem of securing a network against a sequence of intruder attacks through heterogeneous guards. Let a network of agents interacting with each other be represented by a graph G. Let $\mathcal{S}$ be a set of guards, in which every guard $s_i \in \mathcal{S}$ is located on some vertex of G. Every guard has a sensing and response range r_i, such that a guard with a range r_i can detect and respond to an intruder attack on a vertex that is at most r_i hops away from it by marching towards the attacked vertex. The vertex at which a guard s_i is present at time k is described by the map f as

$$f :\ (\mathcal{S}, k)\ \rightarrow\ V. \tag{28}$$

We use $f_k(s_i)$ to denote the vertex where guard s_i is located at time instant k. The notation $f_k(\mathcal{S})$ will be used to denote the set of vertices in G where the guards in $\mathcal{S}$ are present at time k, i.e., $f_k(\mathcal{S}) = \{f_k(s_i) \mid s_i \in \mathcal{S}\}$.

We say that a *vertex* v *is secured* at time k if it lies within a range of at least one

guard at time instant k, i.e.,

$$\exists s_i : \ d(f_k(s_i), v)_G \ \leq \ r_i \tag{29}$$

If (29) is true for all the vertices in G, then G is secured against intruders at time k, and we say that $f_k(\mathcal{S})$ is a *secure configuration* of guards at time k.

In the case of an intruder attack at some vertex $u \in V$, a guard s_i securing u will move from $f_k(s_i)$ to u along the edges of a graph to counter an intruder. This results into a new configuration of guards at time $k+1$, which is given by $f_{k+1}(\mathcal{S})$. If $f_{k+1}(\mathcal{S})$ is also a secure configuration, then the graph remains secure against an intruder attack.

Definition 5.1.1 *Let us assume that a secure configuration of guards at time k, $f_k(\mathcal{S})$, results into another secure configuration, $f_{k+1}(\mathcal{S})$, at time instant $k+1$. The new configuration $f_{k+1}(\mathcal{S})$ is due to the movement of a guard $s_i \in \mathcal{S}$ from a vertex $f_k(s_i)$ to a vertex $f_{k+1}(s_i)$. If this is true for all k, we say that a graph is eternally secure.*

Here we are assuming that at any time instant, there can be an intruder attack only at a single vertex, and only one guard moves to counter this intruder attack. In other words, we are assuming that $\mid f_{k+1}(\mathcal{S}) - f_k(\mathcal{S}) \mid \leq 1$. Therefore, the objective is to investigate the following problem,

How can a graph be eternally secured against a sequence of intruder attacks using a set of guards $\mathcal{S}$, where guards in $\mathcal{S}$ may have different ranges from each other?

Major aspects related to the notion of eternal security in graphs are as follows:

(a) Number of guards required along with their ranges to eternally secure a graph.

(b) Locations of guards on the vertices of a graph, i.e., a secure configuration of guards in a graph.

(c) A strategy for moving guards to counter intruder attacks so that a secure configuration of guards is always maintained.

All of these issues are addressed in the next sections. We present an algorithm that determines a secure configuration for a given a set of guards, and also provides a strategy to maintain a secure configuration in case a guard moves to counter an intruder attack. We also analyze the number of guards required for the eternal security of a graph and how these numbers vary with the ranges of the guards. This analysis allows us to characterize various graph topologies that can be eternally secured with a given number of guards and ranges.

5.2 Eternal Security Through Heterogeneous Guards

In this section, we present a scheme to distribute a given number guards with various ranges among the vertices of a graph to make the graph eternally secure. Let $\mathcal{S} = [s_1, s_2, \cdots, s_\sigma]$ be a vector of given guards, and $\boldsymbol{r} = [r_1, r_2, \cdots, r_\sigma]$ be a vector of their ranges, in which r_i is the range of guard s_i. We begin with the following simple observation.

Lemma 5.2.1 *A single guard with a range r makes a graph G with a diameter d eternally secure if and only if $r \geq d$.*

The above observation provides a systematic way of distributing guards in $\mathcal{S}$ with their corresponding ranges in $\boldsymbol{r}$ to make a graph eternally secure. Our approach is to partition a graph into clusters, and assign a single guard to each cluster. Every guard is then responsible for the security of the nodes in its cluster. The clusters are formed in such a way that the distance between any two nodes of the same cluster is not greater than the range of the guard assigned to the cluster, i.e., $d(u, v)_G \leq r_i$, where u and v are the vertices of the same cluster C_i, and r_i is the range of the guard s_i assigned to C_i. Since the distance between any two nodes in C_i is not greater than r_i, guard s_i is sufficient for the eternal security of all the nodes in C_i by Lemma 5.2.1. A block diagram of the scheme is shown in Fig. 5.1.

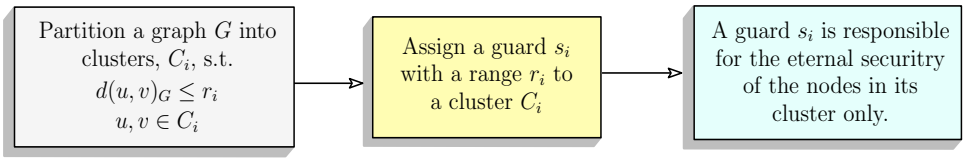


Figure 5.1: A scheme for eternally securing a graph by making clusters.

As an example, consider a graph shown in Fig. 5.2. Let there be three guards, s_1, s_2, and s_3, with ranges $1, 2$, and 3 respectively. The vertices of G are partitioned into three clusters namely C_1, C_2, and C_3. Meanwhile, guards s_1, s_2, and s_3 are assigned to clusters C_1, C_2, and C_3 respectively. It is to be noted that for any cluster C_i, $d(u,v)_G \leq r_i$, $\forall u, v \in C_i$. Therefore, a guard s_i can always detect and respond to an intruder attack on some vertex in C_i.

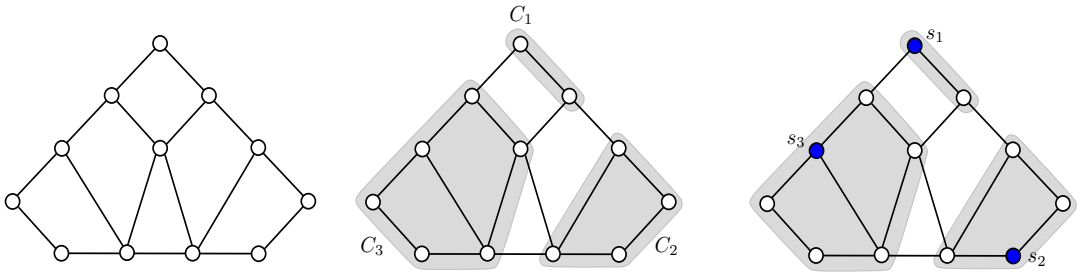


Figure 5.2: An example illustrating the partitioning of graph vertices into clusters for eternal security. Guards s_1, s_2, and s_3, with ranges $1, 2$, and 3 respectively are assigned to clusters C_1, C_2, and C_3.

In a secure configuration of guards within a graph, a node may be secured by more than one guard. In the case of an intruder attack on a node, a response by one of the guards may result into another secure configuration, whereas a response by some other guard may produce a non-secure configuration of guards as shown in Fig. 5.3. Thus, for the eternal security, it is crucial to determine a *right* guard to counter an attack. The clustering approach is particularly useful for that matter as a guard responds to an attack on a vertex in its cluster only. As a result, a secure configuration of guards is always maintained. For a given number of guards and ranges, an effective partitioning of graph vertices into clusters is not a straightforward task. This is the

next topic of our discussion.

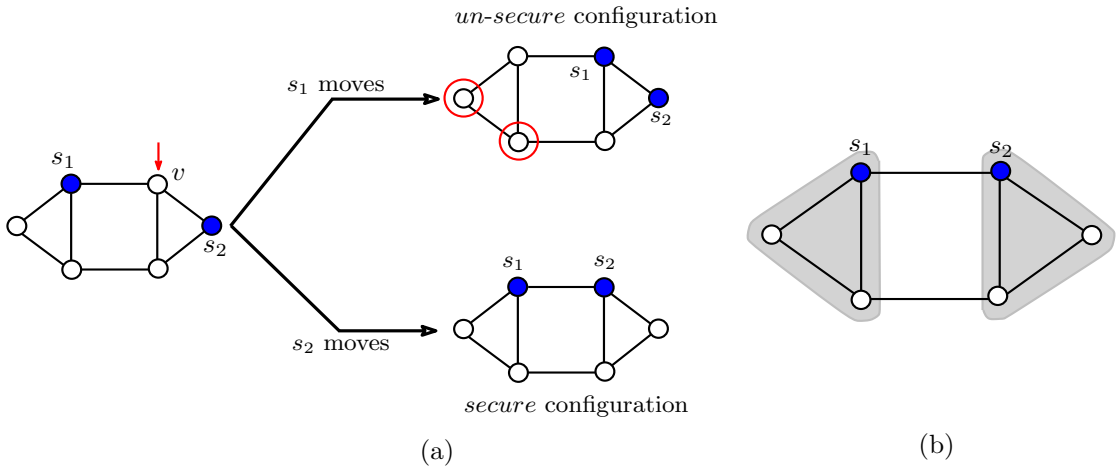


Figure 5.3: (a) Guards s_1 and s_2 are in a secure configuration and both have a range 1. Let there be an attack on a vertex v. Since $d(s_1, v)_G = d(s_2, v)_G = 1$, both guards can counter an attack on v. However, the movement of guard s_1 to v results into an un-secure configuration as the circled nodes are not secured by any guard. On the other hand, the movement of s_2 to counter an attack on v produces a secure configuration. (b) The vertices are partitioned into two clusters, each having a single guard. Each guard is responsible for the security of the vertices in its cluster only.

5.3 Decomposing a Graph into Clusters

The notion of graph power is useful for the decomposition of a graph into clusters for the eternal security of its nodes.

Proposition 5 *A graph G is eternally secured by a single guard with a range r if and only if the corresponding G^r is eternally secured by a guard with a range* 1.

Proof. Let G be a graph with a diameter d, and eternally secured by a single guard with a range r. By Lemma 5.2.1, $r \geq d$, implying that G^r is a complete graph. Since the distance between any two nodes in a complete graph is 1, a single guard with a range 1 is sufficient to eternally secure a graph.

Furthermore, assume G^r to be eternally secured with a single guard of range 1. $u \sim v \in E(G^r)$ implies that $d(u, v)_G \leq r$, for any $u, v \in V$. Thus, a single guard with a range r eternally secures all the vertices in G. ∎

Since a guard with a range r can eternally secure all the vertices in G that lie within a distance r from each other, a guard with a range r can eternally secure all the vertices that induce a complete sub-graph in G^r. Thus, for a guard with a range r, a cluster with the maximum number of vertices can be obtained by selecting all the vertices in a maximum clique, which is obtained by maximal clique decomposition of G^r. As a result, all the vertices in the maximum clique of G^r can be eternally secured by a guard with a range r. Similarly, for another guard with a range r_i, corresponding cluster can be obtained by repeating the same procedure, i.e., obtaining a maximal clique decomposition of G^{r_i} and then selecting a clique with the maximum number of vertices that are not yet eternally secured. This scheme leads us to the algorithm in Section 5.3.1 for obtaining the clusters corresponding to the guards with the given ranges. Before presenting the main algorithm, we define the following routines that will be used in the algorithm.

Max_Clique(H)

This subroutine takes the adjacency matrix of a graph H with n vertices as an input, and performs a maximal clique decomposition of H, returning a matrix $M \in \mathbb{R}^{n \times m}$ with only 0, or 1 entries. Here, m is the total number of maximal cliques. The columns of M contain maximal cliques in such a way that the non-zero entries in the j^{th} column of M indicates vertex indices that constitute the j^{th} maximal clique. Maximal clique decomposition is a well known combinatorial optimization problem. Bron-Kerbosch algorithm [102] is a well known and widely used algorithm for finding maximal cliques in an undirected graph.

Set_Cover(*β_i*, M)

The purpose of this subroutine is to select β_i number of subsets from a given set of subsets in such a way that the cardinality of the union of these β_i subsets is maximized. In terms of the matrix input, this subroutine takes a 0-1 matrix M and a positive integer β_i, and returns the indices of β_i number columns so that performing

the *OR* operation on these columns result into a vector with the maximum number of 1 entries. Set cover problem is a classical and one of the fundamental problems in combinatorics and computer science. A detailed treatment on this topic is provided in [103] along with a number of references.

5.3.1 Main Algorithm

In this sub-section, we present the main algorithm for partitioning a graph into clusters for the eternal security of a graph through a given set of guards along with their ranges. Let α be a vector containing the ranges of guards in a descending order, i.e., the i^{th} element of α, denoted by α_i, is the i^{th} largest range by any guard. Let β be an array where the i^{th} element, denoted by β_i, is the number of guards with the range α_i. For example, $\alpha = [4,\ 2,\ 1]$, and $\beta = [1,\ 3,\ 2]$ indicates that a graph has a single guard with a range 4, three guards with a range 2, and two guards having a range 1. Note that α_1 will be the maximum range by any guard. Furthermore, let V_{cov} be the set of vertices that are included in some cluster $C_{i,j}$, where $C_{i,j}$ is a cluster corresponding to the j^{th} guard with a range i. For instance, if there are three guards each having a range 2, the corresponding clusters will be $C_{2,1}, C_{2,2}$ and $C_{2,3}$.

As an illustration of Algorithm I, consider a network shown in Fig. 5.4. Let us assume that three guards s_1, s_2, and s_3 with ranges $1, 1$, and 3 respectively need to be distributed among the nodes to make the network eternally secure. The objective is to find an initial secure configuration of guards, as well as devise a strategy for the movement of guards so that the guards always maintain a secure configuration. Both of these goals can be achieved by decomposing the graph into clusters, and assigning an appropriate guard to each of the clusters. Using Algorithm I, we begin by arranging the ranges of guards in an array in a descending order, $\alpha = [3\ , 1]$. An array containing the number of guards corresponding to each range in α is $\beta = [1,\ 2]$.

Algorithm I: Decomposing a graph into clusters for the eternal security

Input: $G,\ \alpha,\ \beta$.
Output: Clusters $C_{i,j}$ containing vertices secured by the j^{th} guard with a range i.
Initialization: $i = 1,\ V_{cov} = \emptyset$.
while $i <$ length of α
    $M \leftarrow \mathcal{M}ax\_\mathcal{C}lique(G^{(\alpha_i)})$
    $\tilde{M} \leftarrow M \backslash V_{cov}$ (Deleting the rows in M corresponding to the vertices that are already included in some cluster).
    $c \leftarrow \mathcal{S}et\_\mathcal{C}over(\beta_i, \tilde{M})$
    for $j = 1$ to β_i **do**
        $C_{i,j} \leftarrow$ Set of uncovered vertices determined by the non-zero indices in the c_j^{th} column of M
        $V_{cov} \leftarrow V_{cov} \cup C_{i,j}$
    end for
    $i \leftarrow i + 1$
end while

Since $\alpha_1 = 3$, G^3 is computed. Maximal clique decomposition of G^3 using Bron-Kerbosch algorithm gives

$$\mathcal{M}ax\_\mathcal{C}lique(G^3) = M = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 \end{pmatrix} \begin{matrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ v_7 \\ v_8 \\ v_9 \\ v_{10} \\ v_{11} \\ v_{12} \end{matrix}.$$

Since there is a single guard with a range 3, a clique containing the maximum number of un-secured nodes is selected, and the cluster of nodes for guard s_3 is obtained. All of the nodes are un-secured in the beginning, so the cluster is composed of vertices in a maximum clique in M. As all cliques are maximum here, clique corresponding to the first column ($c = 1$) is picked containing the vertices in $\{v_1, v_2, \cdots, v_8\}$. Thus, s_3 eternally secures the vertices in cluster $C_{3,1} = \{v_1, v_2, \cdots, v_8\}$. This changes V_{cov} from

an empty set to $V_{cov} = \{v_1, v_2, \cdots, v_8\}$. The range of next guard is 1, so maximal clique decomposition of G is obtained as

$$\mathcal{M}ax\_\mathcal{C}lique(G) = M = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{matrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ v_7 \\ v_8 \\ v_9 \\ v_{10} \\ v_{11} \\ v_{12} \end{matrix}.$$

Since vertices in V_{cov} are already secured, a matrix $\tilde{M}$ containing only the unsecured vertices is obtained by deleting the corresponding rows in M.

$$M = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \longrightarrow \tilde{M} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Two columns jointly covering the maximum number of vertices will be selected as there are two guards with a range 1 ($\beta_2 = 2$). Thus, $\mathcal{S}et\_\mathcal{C}over(2, \tilde{M})$ returns $c = [14,\ 15]$. Therefore, the clusters corresponding to the two guards with a range 1 are $C_{1,1} = \{v_9, v_{10}\}$ and $C_{1,2} = \{v_{11}, v_{12}\}$.

Remarks: In our cluster decomposition, for any two vertices $u, v \in C_i$, we have $d(u, v)_G \leq r_i$, in which r_i is the range of the guard assigned to the cluster C_i. Note that the distance considered here is the distance between the vertices in G, that is $d(u, v)_G$, rather than the distance between the vertices in the sub-graph induced by the cluster C_i, that is $d(u, v)_{C_i}$. Since $d(u, v)_G \leq d(u, v)_{C_i}$, a guard with a range less than the diameter of the sub-graph induced by the vertices in C_i may be sufficient

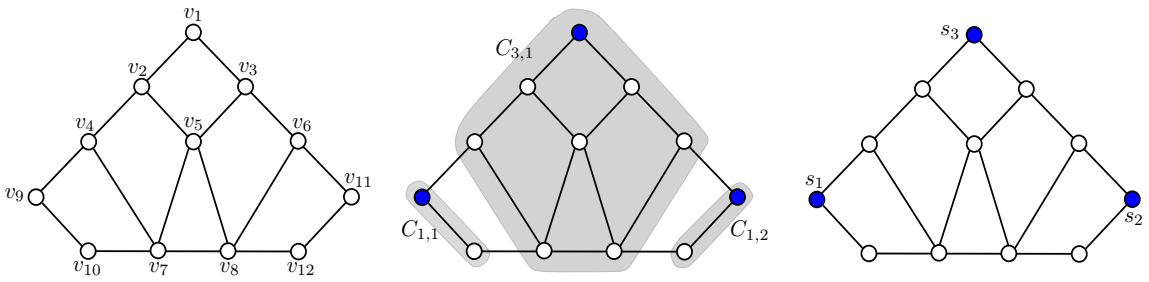
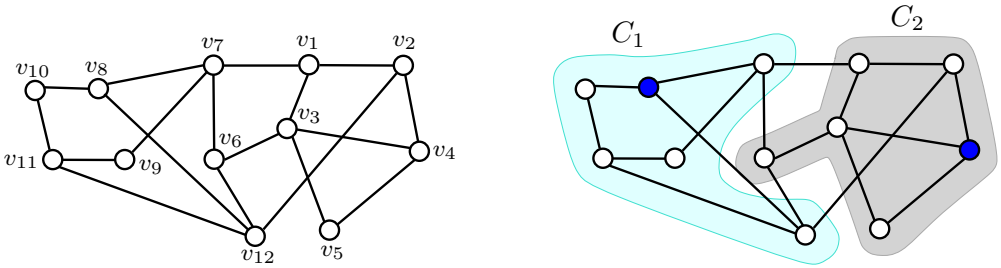


Figure 5.4: A network with twelve nodes. $C_{1,1}$ cluster contains the first guard with range 1, while $C_{1,2}$ contains guard s_2, which is the second guard with range 1. The guard with range 3 is s_3, which secures all the vertices in cluster $C_{3,1}$.

for the eternal security of all the nodes in C_i as shown in the example of Fig. 5.5. This makes our approach better than the one in which a graph is simply divided into induced sub-graphs and for every induced sub-graph, there exists a guard with a range greater than or equal to the diameter of the induced sub-graph.



(a) (b)

Figure 5.5: Two guards, each having a range 2, are available for the eternal security of a given network. In (b), a cluster decomposition using Algorithm I is shown. Since $d(v_2, v_6)_G = 2$, both v_2 and v_6 are included in the same cluster C_2 in spite of the fact that $d(v_2, v_6)_{C_2} = 3$. Moreover, there is no way to divide the graph into two induced sub-graphs in such a way that each sub-graph has a diameter of at most 2. However, it is possible to have two clusters so that the distance between any two nodes of the same cluster is at most 2 as shown in (b).

5.4 Average Distance Moved by a Guard to Counter an Intruder Attack

For the eternal security of a graph, guards with various ranges are located on the vertices of a graph. In the case of an intruder attack on some vertex, these guards

move along the edges of a graph through a path of vertices, covering a certain path length. The vertices of a graph are divided into clusters, $C_1, C_2, \cdots, C_\sigma$, where σ is the total number of guards. All the vertices in a cluster C_i are secured by a single guard with a range r_i, where $r_i \geq \max\limits_{u,v,\in C_i} d(u,v)_G$. Since the maximum distance between any two nodes varies from cluster to cluster, the path lengths covered by guards to counter attacks also vary. Average distance a guard moves to eternally secure a graph may become a significant design parameter for various applications. Thus, we provide an analysis of the average distance moved by a guard to eternally secure a graph by the cluster decomposition. We assume that the probability of a vertex being attacked by an intruder is same for all the vertices of a graph.

Proposition 6 *Let G be a graph whose vertices are decomposed into ℓ clusters, denoted by $C_1, \cdots, C_\ell$, such that $\bigcup\limits_{i=1}^{\ell} C_i = V(G)$. For every cluster C_i, let there be a guard s_i that eternally secures all of the vertices in C_i only. If τ is the average distance (path length) moved by a guard to counter an intruder attack on some vertex $v \in V(G)$, then*

$$\tau = \frac{1}{n}\left[\sum_{i=1}^{\ell} \frac{1}{(n_i - 1)} \sum_{u,v \in C_i} d_G(u,v)\right], \tag{30}$$

where n_i is the number of vertices in the cluster C_i, and n is the total number of vertices in G.

Proof. Let $v \in C_i$, then the average distance between v and some other $u \in C_i$ in G is defined as,

$$\rho(v) = \frac{1}{n_i - 1} \sum_{u \in C_i} d(u,v)_G$$

The average distance between the vertices in C_i, denoted by $\rho(C_i)$, is the average value of the distances between all pairs of vertices in C_i, i.e.,

$$\begin{aligned}\rho(C_i) =& \frac{1}{n_i}\sum_{v\in C_i}\rho(v)\\ =& \frac{1}{n_i(n_i-1)}\sum_{u,v\in C_i} d(u,v)_G.\end{aligned} \tag{31}$$

This is the average distance a guard s_i moves in a cluster C_i to counter an intruder attack on some $v \in C_i$. Since there are ℓ clusters with various number of vertices and guards with various ranges, the average distance a guard moves in response to an attack is the weighted average of $\rho(C_i)$ given as

$$\tau = \frac{1}{n}\sum_{i=1}^{\ell} n_i\rho(C_i). \tag{32}$$

Inserting $\rho(C_i)$ from (31) into (32), we get the desired result,

$$\tau = \frac{1}{n}\sum_{i=1}^{\ell}\frac{1}{(n_i-1)}\sum_{u,v\in C_i} d(u,v)_G.$$

■

For the network in Fig. 5.4, average path length covered by a guard to counter an intruder attack is 1.523.

5.5 A Discrete Event System Model

The phenomenon of eternal security of a graph involves an attack on a vertex that causes the movement of a guard to the attacked vertex. Since a guard is always located on some vertex in a graph, the *state* of a guard (defined as the location of a guard on a vertex) can always be described by a discrete set, i.e., vertex set of the graph. Intruder attack on a vertex is a discrete event as it takes place at some discrete point in time causing a transition in the state of a guard. Thus, the formalism of discrete event systems can be used to model the eternal security in graphs through a set of heterogeneous guards.

A discrete event system is frequently represented by an *automaton* (e.g., [104]), which can be written as

$$\mathcal{G} = (X, E, f, \Gamma, x_0, X_m),$$

where

X is the set of *states*,

E is the set of *events*,

$f : X \times E \rightarrow X$ is the *transition function*,

$\Gamma : X \rightarrow 2^E$ is the *active event function*, i.e, $\Gamma(x) \subseteq E$ is the set of events for which $f(x, e)$ is defined,

x_0 is the initial state, and

$X_m \subseteq X$ is the set of marked states.

For our problem of eternal security, we define an atomic discrete event system from which the overall system will be derived. In fact, the atomic system is defined for a single guard s_i with range $r_i \geq 1$, and sub-graph $G_{(i)}$ with vertex set $V_{(i)}$ in such a way that $d(u, v) \leq r_i$, $\forall\ u, v \in V_{(i)}$. The *state* of the atomic discrete event system is the location of the guard on some vertex $v_j \in V_{(i)}$. An event e_k is defined as an intruder attack on a vertex $v_k \in V_{(i)}$. Automaton for this system is given by

$$\mathcal{G}_{(i)} = \left(V_{(i)}, E_{(i)}, f_{(i)}, \Gamma_{(i)}\right), \tag{33}$$

where

$V_{(i)}$ is the vertex set of $G_{(i)}$, and

$E_{(i)} = \{e_k : v_k \in V_{(i)}\}$. Here e_k is the intruder attack on vertex v_k.

$f_{(i)} : V_{(i)} \times E_{(i)} \rightarrow V_{(i)}$ is the transition function, where $f_{(i)}(v_j, e_k) = v_k$, i.e., in case of

an attack on a vertex v_k, guard g_i lying on a vertex v_j will move to vertex v_k.

$\Gamma(v_j) = E_{(i)}$, $\forall v_j \in V_{(i)}$.

States are marked when it is desired to give them a special meaning. For instance, these states may be the ones that are reached after a certain sequence of events. Since there are no marked states for our system, we can ignore them in our model.

Furthermore, consider a graph G having a vertex set V. Instead of having a single guard, let there be σ guards, $s_1, s_2, \cdots, s_\sigma$ having ranges $r_1, r_2, \cdots, r_\sigma$ respectively. The goal is to make G eternally secure, and obtain a corresponding discrete event system model. We use the same approach as in Section 5.2, i.e., to decompose a graph into clusters and make each guard responsible for the eternal security of the vertices in its cluster only. The discrete event system corresponding to each guard s_i, and its cluster can be described in a similar way as in (33). Since there are σ guards, there will be σ such systems, each having its own state space, $V_{(i)}$. Note that $\bigcup_{i=1}^{\sigma} V_{(i)} = V$, and $V_{(i)} \cap V_{(j)} = \emptyset$, $\forall i \neq j$. The discrete event system $\mathcal{G}$, which corresponds to the overall graph G with guards $s_1, s_2, \cdots, s_\sigma$, can be described by the *parallel compositions* of automata $\mathcal{G}_{(i)}$, $\forall i \in \{1, 2, \cdots, \sigma\}$. The parallel composition of two atomic automata $G_{(1)}$, and $G_{(2)}$ is defined here as[1]

$$G_{(1)} \parallel G_{(2)} := \big(V_{(1)} \times V_{(2)},\ E_{(1)} \times E_{(2)},\ f,\ \Gamma_{1\parallel 2},\ (x_{0,1}, x_{0,2})\big)\,, \tag{34}$$

where

$$f((v_i, v_j), e) \ := \ \begin{cases} \big(f_{(1)}(v_i, e),\ v_j\big) & \text{if } e \in \Gamma_{(1)}(v_i) \\ \big(v_i,\ f_{(2)}(v_j, e)\big) & \text{if } e \in \Gamma_{(2)}(v_j), \end{cases}$$

and $\Gamma_{1\parallel 2} = \Gamma_{(1)}(v_i) \cup \Gamma_{(2)}(v_j)$.

Using the associativity property of parallel compositions of automata [104], the required discrete time system for the eternal security of graph G with guards $s_1, s_2, \cdots, s_\sigma$ is given by

[1]Here we are making use of the fact that there are no common events between automata $G_{(1)}$, and $G_{(2)}$, i.e., $E_{(1)} \cap E_{(2)} = \emptyset$. This is true as $V_{(1)} \cap V_{(2)} = \emptyset$.

$$\mathcal{G} = \mathcal{G}_{(1)} \parallel \mathcal{G}_{(2)} \parallel \cdots \mathcal{G}_{(\sigma)}. \tag{35}$$

It is to be mentioned here that the state space $V_{(i)}$ of an automaton $G_{(i)}$ corresponds to the vertices that need to be secured by the guard s_i with range r_i, and can be obtained by the cluster decomposition method outlined in Section 5.3.

As an example, consider a graph G shown in Fig. 5.6(a). The graph has two guards s_1 and s_2 with ranges 2 and 1 respectively. The vertex set of G is decomposed into two clusters $V_{(1)} = \{v_1, v_2, v_3, v_4\}$ and $V_{(2)} = \{v_5, v_6\}$. Guard s_1 is responsible for the eternal security of the vertices in $V_{(1)}$, whereas s_2 secures the vertices in $V_{(2)}$. The discrete event system corresponding to s_1 and $V_{(1)}$ is given by an automaton $\mathcal{G}_{(1)} = \big(V_{(1)}, E_{(1)}, f_{(1)}, \Gamma_{(1)}\big)$ in which $E_{(1)} = \{e_1, e_2, e_3, e_4\}$ is an event set, and $e_i \in E_{(1)}$ means that there is an attack on vertex v_i. The state transition function is given by $f_{(1)}(v_i, e_j) = v_j$, which means that guard s_1 moves from its current location at vertex v_i to vertex v_j in case of an attack at v_j. Similarly, discrete event system corresponding to guard s_2 is represented by $\mathcal{G}_{(2)} = \big(V_{(2)}, E_{(2)}, f_{(2)}, \Gamma_{(2)}\big)$. The state transition diagrams for both $\mathcal{G}_{(1)}$, and $\mathcal{G}_{(2)}$ are shown in Fig. 5.6.

Using the formulation in (34), discrete event system $\mathcal{G}$, which corresponds to the overall graph G, is obtained by the parallel combination of $\mathcal{G}_{(1)}$ and $\mathcal{G}_{(2)}$. Fig. 5.7 shows the state transition diagram of $\mathcal{G}$.

5.6 Eternal Security Through Homogeneous Guards

Eternal security in graphs through homogeneous guards[2] can be achieved by adapting the same approach, i.e., partitioning of graph vertices into clusters and assigning a guard to each cluster. For k number of guards, where each guard has a range m, clusters can be obtained by the steps as follows:

a) Take the r^{th} power of a given graph, G^r.

[2]Homogeneous guards are the ones that have a same range $r \geq 1$.

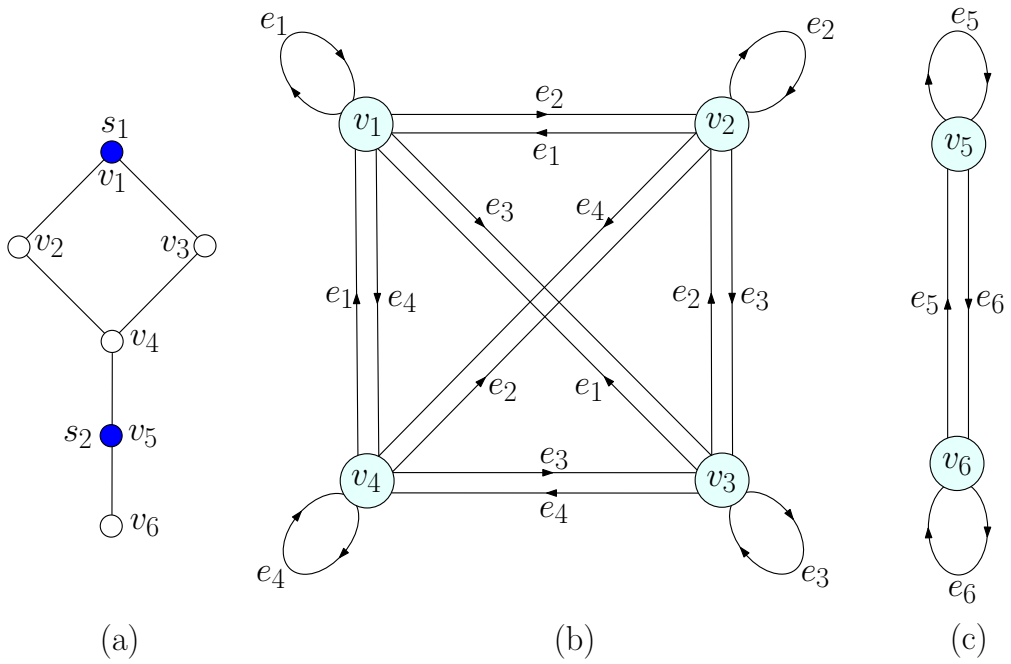


Figure 5.6: (a) A graph G containing two guards s_1, and s_2. (b) The state transition diagram of $\mathcal{G}_{(1)}$. (c) The state transition diagram of $\mathcal{G}_{(2)}$.

b) Obtain a maximal clique decomposition of G^r. Let there be m maximal cliques.

c) Pick k maximal cliques that jointly cover the maximum number of vertices in the graph.

d) Each maximal clique is a cluster. Assign a single guard to the vertices in a cluster.

The issue of minimum number of guards required to eternally secure a graph needs to be addressed. In this section, we provide tight bounds on the minimum number of guards required to eternally secure a graph when all guards have a same range $r \geq 1$. We begin by defining the eternal security number of a graph with homogeneous guards.

Definition 5.6.1 *The eternal security number of a graph G, denoted by $\sigma_r(G)$, is the minimum number of guards required to eternally secure a graph G when all the guards have a same range r.*

In [84], fundamental lower and upper bounds are presented for the eternal security

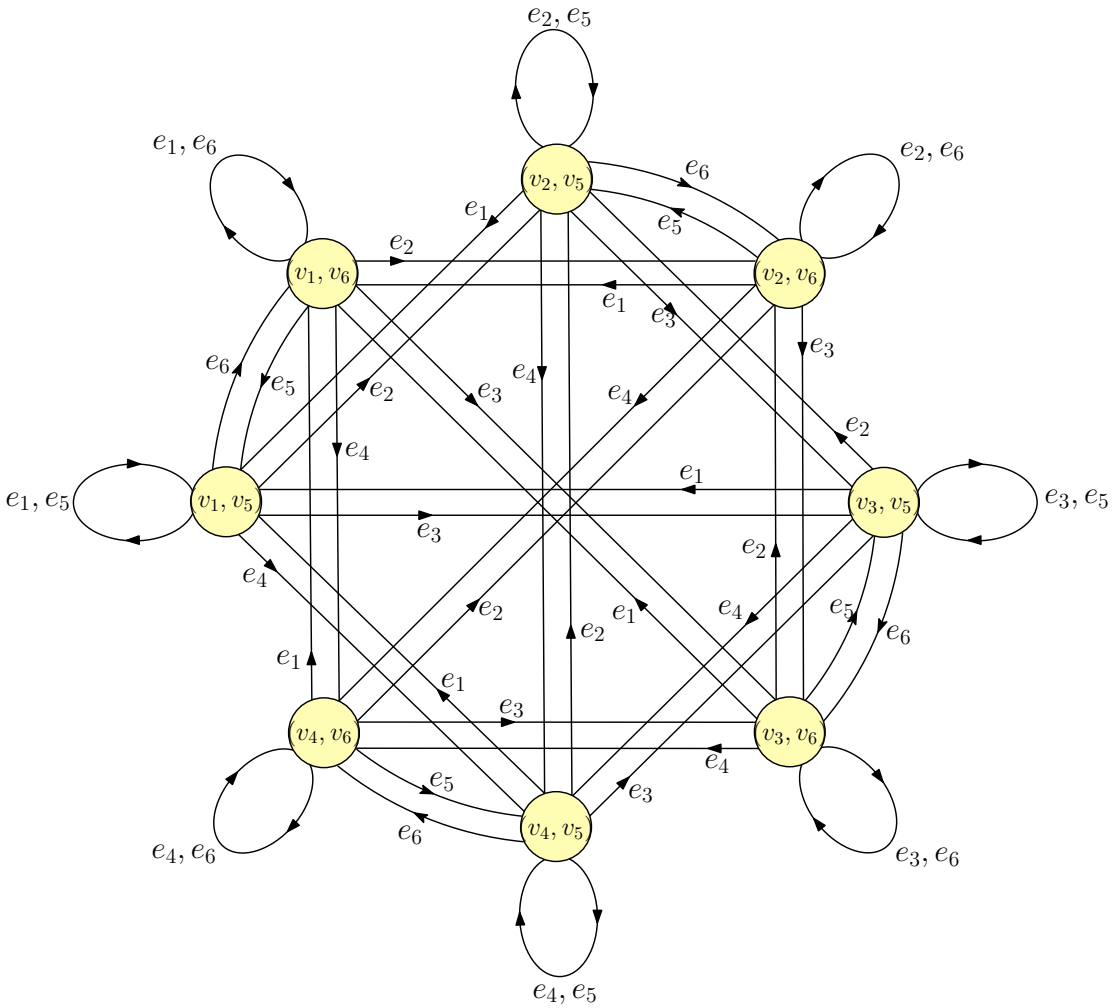


Figure 5.7: The state transition diagram of $\mathcal{G} = \mathcal{G}_{(1)} \parallel \mathcal{G}_{(2)}$.

number of a graph for $r = 1$ (i.e., when all the guards can detect and respond to an attack on their immediate neighbors only). The bounds relate $\sigma_1(G)$ to the independence number $\iota(G)$, and the clique cover number $\theta(G)$ of a graph G. Clique cover number is the minimum number of cliques needed to cover all the vertices of G.

Theorem 5.6.1 *[84] For any graph G,*

$$\iota(G) \ \leq \ \sigma_1(G) \ \leq \theta(G). \tag{36}$$

Many nice results are available regarding $\iota(G)$ and $\theta(G)$ that can be directly used for computing $\sigma_1(G)$. In a similar way, relating $\sigma_r(G)$ (for $r > 1$) to $\sigma_1(G)$ is useful.

Lemma 5.6.1 *For a graph G,*

$$\sigma_r(G) \;=\; \sigma_1(G^r). \tag{37}$$

Proof. Let G be eternally secured with x number of guards, where each guard has a range r, so that for any time instant k and vertex $v \in V$, there exists a guard s such that $d(f_k(s), v)_G \leq r$. By the definition of graph power G^r, $d(f_k(s), v)_G \leq r \Leftrightarrow d(f_k(s), v)_{G^r} \leq 1$.[3] Thus, the same number of guards x, where each guard has a range 1, can eternally secure G^r, i.e., $\sigma_1(G^r) = x$.

By a similar argument it can be shown that $(\sigma_1(G^r) = x) \Rightarrow (\sigma_r(G) = x)$, which gives the required relation. ∎

A general form of (37) is expressed in the following result.

Theorem 5.6.2 *For any graph G, and positive integers m and n,*

$$\sigma_m(G^n) \;=\; \sigma_n(G^m). \tag{38}$$

Proof. Let $G^n = Y$, and $G^m = Z$. Using (37), $\sigma_m(Y) = \sigma_1(Y^m) = \sigma_1(G^{mn})$. The right side of (38) is $\sigma_n(Z) = \sigma_1(Z^n) = \sigma_1(G^{mn})$, giving the required result. ∎

Furthermore, we utilize the notion of *domination* in graphs for finding sharp bounds on the eternal security number of graphs. The concept of domination in graphs is closely related to the idea of eternal security. An r-distance dominating set, or simply an *r-dominating set* of a graph, denoted by $X^{(r)} \in V(G)$, is a subset of vertices such that for each $v \in V(G)$, either $v \in X^{(r)}$, or v is at most r distant from some vertex in $X^{(r)}$. The cardinality of a minimum r-dominating set is the r-domination number of a graph, denoted by $\gamma^{(r)}(G)$. We use the notation $\mathcal{N}_r(v)$ to denote the open r-neighborhood of a vertex v, i.e., $\mathcal{N}_r(v) = \{u \in V(G) \mid d(u, v)_G \leq r\}$. Similarly, the closed r-neighborhood of a vertex v, denoted by $\mathcal{N}_r[v]$, is $\mathcal{N}_r(v) \cup \{v\}$.

[3] $f_k(s)$ is the vertex, where guard s is located at time k.

Theorem 5.6.3 *For any graph G,*

$$\sigma_{2r}(G) \leq \gamma^{(r)}(G), \tag{39}$$

where $\gamma^{(r)}$ is the r-domination number of G.

Proof. Let $X^{(r)} = \{x_1, x_2, \cdots, x_{\gamma^{(r)}}\}$ be a minimum r-dominating set of G. Let G_{x_i} be a sub-graph induced by the vertices in $\mathcal{N}_r[x_i]$. By the definition of an r-dominating set, $d(v, x_i)_{G_{x_i}} \leq r$, $\forall v \in \mathcal{N}_r[x_i]$. Thus, for any $y, z \in \mathcal{N}_r[x_i]$, $d(y, z)_{G_{x_i}} \leq 2r$, implying that $diam(G_{x_i}) \leq 2r$. By Lemma 5.2.1, $\sigma_1^{(2r)}(G_{x_i}) = 1$. This is true for each $x_i \in X^{(r)}$. Since $\bigcup_{x_i \in X^{(k)}} G_{x_i} \subseteq G$, we get $\sigma_{2r}(G) \leq \gamma^{(r)}$. ∎

For $r = 1$, we have $\sigma_2(G) \leq \gamma(G)$, where $\gamma(G)$ is a 1-domination number (or simply a domination number) of a graph. An example illustrating the above proof for $r = 1$ is shown in the Fig. 5.8. It is to be mentioned here that the bound in (39) is tight. For example, consider the graph in Fig. 5.8, where $\sigma_2(G) = \gamma(G) = 2$.

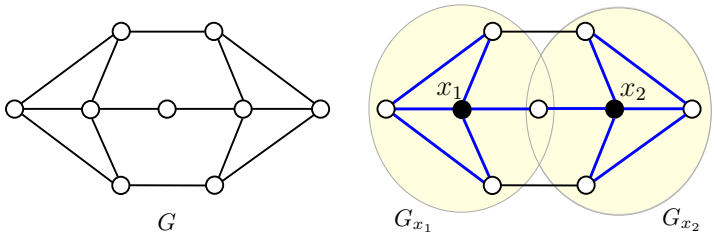


Figure 5.8: $X = \{x_1, x_2\}$ is a dominating set of a given graph G. For each $x_i \in X$, there exits a sub-graph G_{x_i} with $diam(G_{x_i}) = 2$, so $\sigma_2(G_{x_i}) = 1$. At the same time, $G_{x_1} \cup G_{x_2} \subseteq G$, so $\sigma_2(G) \leq [\sigma_2(G_{x_1}) + \sigma_2(G_{x_2})] = 2$.

Connected Domination Number and $\sigma_r(G)$

We also relate $\sigma_r(G)$ to a widely studied notion of connected domination in graphs. For a connected graph, a *connected dominating set*, X_c, is a dominating set in which the vertices in X_c induce a connected subgraph. The *connected domination number*, denoted by γ_c, is the cardinality of a minimum connected dominating set.

Theorem 5.6.4 *Let G be a connected graph, and G_c be a sub-graph induced by the vertices in a minimum connected dominating set of G. If G_c is eternally secured by*

a certain number of guards each having a range r, then G is eternally secured by the same number of guards each having a range $r+2$.

Proof. Let X_c be a minimum connected dominating set, and G_c be a sub-graph induced by the vertices in X_c. Let G_c be eternally secured by σ number of guards, where each guard has a range r. It means that for any time instant k and $x \in X_c$, there exists a guard s such that $d(f_k(s), x)_{G_c} \leq r$, in which $f_k(s)$ is the vertex where a guard s is located at time k.

If $x \in X_c$, then $d(u, v)_G \leq 2$, $\forall u, v \in \mathcal{N}[x]$. It implies that there always a guard s such that $d(f_k(s), u)_G \leq r+2$, $\forall u \in \mathcal{N}[x]$. Since every vertex in G lies in the closed neighborhood of some vertex in X_c, all vertices in G are eternally secured by the σ number guards, where each guard has range $r+2$. ∎

5.7 Eternal Security Number for Some Classes of Graphs

In this section, we give explicit expressions and bounds for $\sigma_r(G)$ for different classes of graphs. We start with a path graph.

Theorem 5.7.1 *Let P_n be a path graph with n vertices, then*

$$\sigma_r(P_n) = \left\lceil \frac{n}{r+1} \right\rceil .$$

Proof. The chromatic number[4] of the r^{th} power of a path graph with n vertices is $r+1$, i.e., $\chi(P_n^r) = r+1$ [105]. Moreover, $\iota(G) \geq \frac{n}{\chi(G)}$, in which $\iota(G)$ and $\chi(G)$ are the independence number and the chromatic number of a graph G respectively [105]. Thus, we get $\iota(P_n^r) \geq \frac{n}{r+1}$, and (36) implies that $\sigma_1(P_n^r) \geq \lceil \frac{n}{r+1} \rceil$. At the same time, every $(r+1)$ consecutive vertices in P_n^r induce a complete sub-graph. Thus, we get $\lceil \frac{n}{r+1} \rceil$ cliques, implying that $\sigma_1(P_n^r) \leq \lceil \frac{n}{r+1} \rceil$. Observing that $\sigma_1(P_n^r) = \sigma_r(P_n)$, we get the desired result. ∎

[4]The chromatic number of a graph G, denoted by $\chi(G)$, is the smallest number of colors needed to color the vertices so that no two adjacent vertices share the same color.

Eternal security number of a cycle graph is of particular interest. Consider a closed boundary of some region, where the boundary has n potential attack points that need to be secured with a certain number of guards. For a given sensing and response range of guards, the eternal security number $\sigma_r(C_n)$ gives the minimum number of guards needed to secure a region. At the same time, for a given number of guards, we can determine the minimum range each guard should have to eternally secure a graph.

Theorem 5.7.2 *Let C_n be a cycle graph with n nodes, then*

$$\left\lfloor \frac{n}{r+1} \right\rfloor \leq \sigma_r(C_n) \leq \left\lceil \frac{n}{r+1} \right\rceil.$$

Proof. From [106], we know that $\iota(C_n^r) = \lfloor \frac{n}{r+1} \rfloor$. Using this result along with (36) and (37), we get $\sigma_r(C_n) \geq \lfloor \frac{n}{r+1} \rfloor$. Moreover, assume that vertices of C_n are labelled consecutively $\{1, 2, \cdots, n\}$. Consider a partition $\mathcal{P}$ of $V(C_n^r)$ as

$$\mathcal{P} = \{\{1, \cdots, r+1\},\ \{r+2, r+3, \cdots, 2(r+1)\}, \cdots, \{x, x+1, \cdots, n\}\},$$

where $x = \left[\left(\lceil \frac{n}{r+1} \rceil - 1\right)(r+1) + 1\right]$. Note that all the vertices in each subset of $\mathcal{P}$ are adjacent to each other in C_n^r. Thus, the vertices in each subset of $\mathcal{P}$ induce a clique in C_n^r. The cardinality of $\mathcal{P}$ is $\lceil \frac{n}{r+1} \rceil$. This gives the clique cover number of C_n^r as $\theta(C_n^r) = \lceil \frac{n}{r+1} \rceil$. Using (36), we directly imply that $\sigma_r(C_n) \leq \lceil \frac{n}{r+1} \rceil$. ∎

Following result is a direct consequence of Theorem 5.7.2.

Corollary 5.7.1 *Every hamiltonian graph*[5] *has*

$$\sigma_r(G) \leq \left\lceil \frac{n}{r+1} \right\rceil.$$

Another useful result regarding 2-connected graphs[6] directly follows from Theorem 5.7.2, and Corollary 5.7.1.

[5] *A hamiltonian cycle in a graph G is a cycle that passes through each vertex exactly once. A graph containing such a cycle is a hamiltonian graph.*

[6] A graph is 2-connected if there does not exist a single vertex whose removal disconnects the graph.

Corollary 5.7.2 *For $r \geq 2$, every 2-connected graph has*

$$\sigma_r(G) \leq \left\lceil \frac{n}{r+1} \right\rceil.$$

Proof. If G is 2-connected, then G^2 is hamiltonian [107]. The required result is then directly implied from Corollary 5.7.1. ∎

Definition 5.7.1 *(Cartesian Product of Graphs): The Cartesian product of graphs G and H, denoted by $G\Box H$, is a graph such that*

(a) the vertex set of $(G\Box H)$ is the Cartesian product $V(G) \times V(H)$.

(b) any two vertices (u, u') and (v, v') are adjacent in $G\Box H$ if and only if $u = v$, and u' is adjacent to v' in H; or $u' = v'$, and u is adjacent to v in G.

Cartesian product of graphs is useful in representing the network topologies that appear in numerous practical scenarios, including grid graph, which is a Cartesian product of two path graphs $P_n\Box P_m$; and prism graph, which is a Cartesian product of a cycle and a path graph $C_n\Box P_m$. These graph structures are useful in designing secure systems as they provide various levels of security in a systematic way. For example, consider a $C_6\Box P_4$ shown in Fig. 5.9. The region enclosed by the innermost circle is protected by four circles, which may correspond to four levels (or layers) of security.

We prove the following lemma that will be used later.

Lemma 5.7.1

$$\sigma_r(P_n\Box P_2) \leq \left\lceil \frac{n}{r} \right\rceil.$$

Proof. Let the vertices of P_n be labelled consecutively as $\{1, 2, \cdots, n\}$, and vertices of P_2 be labelled as $\{1', 2'\}$. The vertex set of $P_n\Box P_2$ can be written as

$$V(P_n\Box P_2) = \{(1,1'), (2,1'), \cdots, (n,1'), (1,2'), (2,2'), \cdots, (n,2')\}.$$

Let us do the partion of $V(P_n \Box P_2)$ as

$$\begin{aligned}\mathcal{P} =&\{\{(1,1'),\cdots,(r,1'),(1,2'),\cdots,(r,2')\},\\ &\{(r+1,1'),\cdots,(2r,1'),(r+1,2'),\cdots,(2r,2')\},\\ &\cdots,\{(x,1'),(x+1,1'),\cdots,(n-1,1'),(n,1'),\\ &(x,2'),(x+1,2'),\cdots,(n-1,2'),(n,2')\}\},\end{aligned}$$

where $x = \left(\lceil \frac{n}{r} \rceil - 1\right)(r) + 1$.

Note that the vertices in each subset of $\mathcal{P}$ induce a complete sub-graph in $(P_n \Box P_2)^r$. Also the cardinality of $\mathcal{P}$ is $\lceil \frac{n}{r} \rceil$. Thus, $\theta((P_n \Box P_2)^r) \leq \lceil \frac{n}{r} \rceil$, and therefore, $\sigma_r(P_n \Box P_2) \leq \lceil \frac{n}{r} \rceil$. ■

Lemma 5.7.1 is useful for obtaining bounds on σ_r of prism $(C_n \Box P_m)$, and grid graphs $(P_n \Box P_m)$, as these graphs can be decomposed into connected components, where each component is a $(P_n \Box P_2)$.

Theorem 5.7.3 *For an even n,*

$$\sigma_r(C_n \Box P_m) \leq \frac{n}{2} \left\lceil \frac{m}{r} \right\rceil. \tag{40}$$

Proof. Let the consecutive vertices of C_n be labelled as $\{1, 2, \cdots, n\}$, and the consecutive vertices of P_n as $\{1', \cdots, m'\}$. The vertex set of $C_n \Box P_n$ can be written as

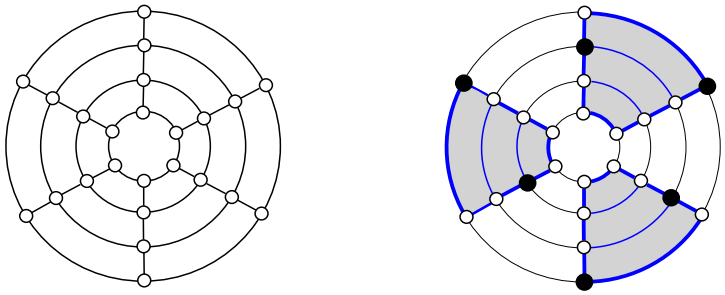
$$\begin{aligned}V(C_n \Box P_m) =&\{(1,1'),(2,1'),\cdots,(n,1'),\\ &(1,2'),(2,2')\cdots,(n,2'),\\ &\cdots,(1,m'),(2,m'),\cdots,(n,m')\}.\end{aligned}$$

Let us do the partition of $V(C_n \Box P_m)$ as

$$\begin{aligned}\mathcal{P} =&\{\{(1,1'),\cdots,(1,m'),(2,1'),\cdots(2,m')\}\\ &\{(3,1'),\cdots,(3,m'),(4,1'),\cdots(4,m')\}\\ &\cdots,\{(n-1,1'),\cdots,(n-1,m'),(n,1'),\cdots,(n,m')\}\}.\end{aligned}$$

Each subset in $\mathcal{P}$ induces a $P_m \Box P_2$. The cardinality of $\mathcal{P}$ is $n/2$. Using Lemma 5.7.1, we get $\sigma_r(C_n \Box P_m) \leq \frac{n}{2} \left\lceil \frac{m}{r} \right\rceil$. ∎

An example in Fig. 5.9 illustrates the above proof.



(a) (b)

Figure 5.9: (a) A graph $(C_6 \Box P_4)$. (b) There are three copies of sub-graph $(P_4 \Box P_2)$ in $(C_6 \Box P_4)$. Since two guards, where each guard has range 2, are sufficient for the eternal security of $(P_4 \Box P_2)$, we get $\sigma_2(C_6 \Box P_4) \leq 6$.

If n is an odd number in $(C_n \Box P_m)$, we can always decompose $(C_n \Box P_m)$ into $(C_{n-1} \Box P_m)$, and C_n, such that is $((C_{n-1} \Box P_m) \cup C_n) \subset (C_n \Box P_m)$. Using Theorem 5.7.3, $\sigma_r(C_{n-1} \Box P_m) \leq \left(\frac{n-1}{2}\right) \left\lceil \frac{m}{r} \right\rceil$, and from Theorem 5.7.2, $\sigma_r \leq \left\lceil \frac{n}{r+1} \right\rceil$. Thus, for an odd n,

$$\sigma_r(C_n \Box P_m) \leq \left(\frac{n-1}{2}\right) \left\lceil \frac{m}{r} \right\rceil + \left\lceil \frac{n}{r+1} \right\rceil . \tag{41}$$

The bound in Theorem 5.7.3 is tight. For example, consider $(C_4 \Box P_3)$, which has a diameter 4. Using Lemma 5.2.1, $\sigma_3(C_4 \Box P_3)$ is at least 2, which is the same value as obtained from (40).

Theorem 5.7.4 *For an even n, a grid graph $(P_n \Box P_m)$ has*

$$\sigma_r(P_n \Box P_m) \leq \frac{n}{2} \left\lceil \frac{m}{r} \right\rceil .$$

Proof. The proof is exactly same as that of Theorem 5.7.3 as $P_n \Box P_m$ can be decomposed into $\frac{n}{2}$ vertex disjoint copies of $P_2 \Box P_m$. ∎

When n is odd, we can decompose $(P_n \Box P_m)$ into $(P_{n-1} \Box P_m)$, and P_n, such that $((P_{n-1} \Box P_m) \cup P_n) \subset (P_n \Box P_m)$. Using Theorem 5.7.4, and Theorem 5.7.1, we get

$$\sigma_r(P_n \Box P_m) \leq \left(\frac{n-1}{2}\right) \left\lceil \frac{m}{r} \right\rceil + \left\lceil \frac{n}{r+1} \right\rceil . \tag{42}$$

In this chapter, we investigated the problem of securing a network of agents against a sequence of intruder attacks by modeling the network as a graph. A certain number of guards, which are capable of detecting and responding to an attack within a certain distance, were deployed throughout the network. The location of guards, and their movement strategies were developed to eternally secure a graph in the sense that in case of an attack on some vertex, there always existed a guard to counter the attack by moving towards the attacked vertex. We extended the idea of eternal security in graphs by incorporating the notion of heterogeneous guards. It is shown that the problem of eternal security in graphs can be resolved efficiently by decomposing a graph into clusters, and assigning a single guard to each cluster. We addressed all aspects of the problem, including number of guards required, location of guards within a graph, and movement strategies of guards to counter any intruder attack. A discrete event system model of the eternal security in graphs is also presented to approach this issue from a system theoretic view-point. Various bounds on the number of guards required to eternally secure a graph are provided for different classes of graphs. The framework provided can be used to analyze as well as design secure network topologies for various applications.

CHAPTER VI

ROBUST GRAPH TOPOLOGIES FOR NETWORKS

Robustness of networked systems against noise corruption and structural changes in an underlying network topology is a critical issue for a reliable performance. In this chapter, we investigate this issue of robustness in networked systems both from structural and functional viewpoints. Structural robustness deals with effects of changes in a graph structure due to link or edge failures, whereas functional robustness addresses how well a system behaves in the presence of noise. We discuss that both of these aspects are very much inter-related, and can be measured through a common graph invariant. A graph process is introduced, in which edges are added to a graph in a step-wise manner to maximize robustness. Moreover, a relationship between the symmetry of an underlying network structure and robustness is also discussed.

6.1 Robustness Issues in Networked Systems

In distributed systems, agents exchange information with each other through local interactions. These interactions in turn define an information exchange network that can be modeled by a graph. For example, in agreement and consensus related problems, agents are required to agree on a common value (that may be a sensor measurement) by implementing a linear consensus protocol. In fact, connectivity of the underlying graph structure is a necessary requirement for the consensus protocol to work. Moreover, structure of the underlying network affects various properties of the system including convergence rates, connectivity of the network under edge (interconnection among agents) or vertex (agent) failures. A highly connected network is obviously less affected by an edge or vertex failures and is therefore, more *robust* to these deletions. Thus, structure of the inter-connection infrastructure plays a key role

in understanding the effects of edge or vertex failures.

Another aspect of robustness comes into play when we also consider agents' dynamics in such systems. Agents compute states (that may be their positions or any other measurements) and eventually exchange them with others through some medium that may be noisy. This noise plays an important role in determining the overall functionality of the system. It has been observed that some network topologies are least affected by the incorporation of noise when agents are performing linear consensus, while others are affected to a larger extent (see [92] for example). The network structures minimally affected by noise are obviously more *robust*. This leads us towards the study of two aspects of *robustness* in multiagent systems, in which agents implement consensus protocols.

(a) *Structural Robustness:* It is the ability of the network to maintain its original structure and connectedness among vertices in the underlying graph under edge or vertex failures.

(b) *Functional Robustness:* It measures how well a system behaves in the presence of noise that corrupts measurements or an information exchange among agents.

The above mentioned robustness views seem to have a different focus, where (a) is related purely to a property of the underlying graph structure while (b) deals with the effect of noise on measurements and states of the agents. We show here that both these robustness views are in fact, related to each other and can be measured by the same parameter.

6.1.1 Structural Robustness vs. Functional Robustness

There may exist multiple paths between two nodes in a given graph of a network. A large number of unique paths between two nodes implies that these nodes are highly interconnected with each other. Thus, their connectivity with each other will not be effected to a large extent by an edge failure, indicating a robust connection between

these nodes. The number of unique paths between any two nodes, therefore, hints upon the quantitative aspect of structural robustness in a network.

It is not only the number of unique paths, but also the *quality* of paths that is crucial to the robustness against edge failures. A path of shorter length between two nodes is preferred over a longer one as it corresponds to an increased level of connectedness between these nodes due to a lesser delay. Also, shorter length paths between nodes result in short random walks that are less affected by the node or edge failures as shown in [121].

Thus, the structural robustness should incorporate both the quantitative as well as the qualitative effect of edge removals on the overall connectivity of the network. As it is shown in [90], the notion of *effective resistance* between two nodes takes into account both of these aspects, i.e., the number of paths between two nodes and the length of these paths. Effective resistance between nodes decreases with an increase in the number of paths between nodes. Moreover, effective resistance between nodes is smaller if the length of the paths between them is shorter. This provides a nice way to quantify the structural robustness in networks.

The *effective resistance*, $r_{i,j}$, between vertices i and j in an un-weighted graph G is defined as the effective electrical resistance between points i and j when a resistor of unit resistance is placed along every edge and a potential difference is applied between i and j as illustrated in the Fig. 6.2. Consider a network in the Fig. 6.2. There are three unique paths between 1 and 2, namely $x = [1 \rightarrow 2]$, $y = [1 \rightarrow 3 \rightarrow 4 \rightarrow 2]$ and $z = [1 \rightarrow 3 \rightarrow 5 \rightarrow 7 \rightarrow 8 \rightarrow 9 \rightarrow 6 \rightarrow 4 \rightarrow 2]$. Each of these paths adds to the robustness of connection between nodes 1 and 2. Since path z is the longest one, it offers least contribution towards the robustness of inter-connection between nodes 1 and 2. This is also indicated by only a slight increase in the $r_{1,2}$ value in Fig. 6.2(c), where the loss of an edge $5 \sim 7$ results in the loss of z path between 1 and 2. Similarly, when a path y is lost, $r_{1,2}$ is increased in greater amount as y path has a

smaller length than z. When a shortest (most crucial) path, x, is lost, the greatest increase in $r_{1,2}$ is observed.

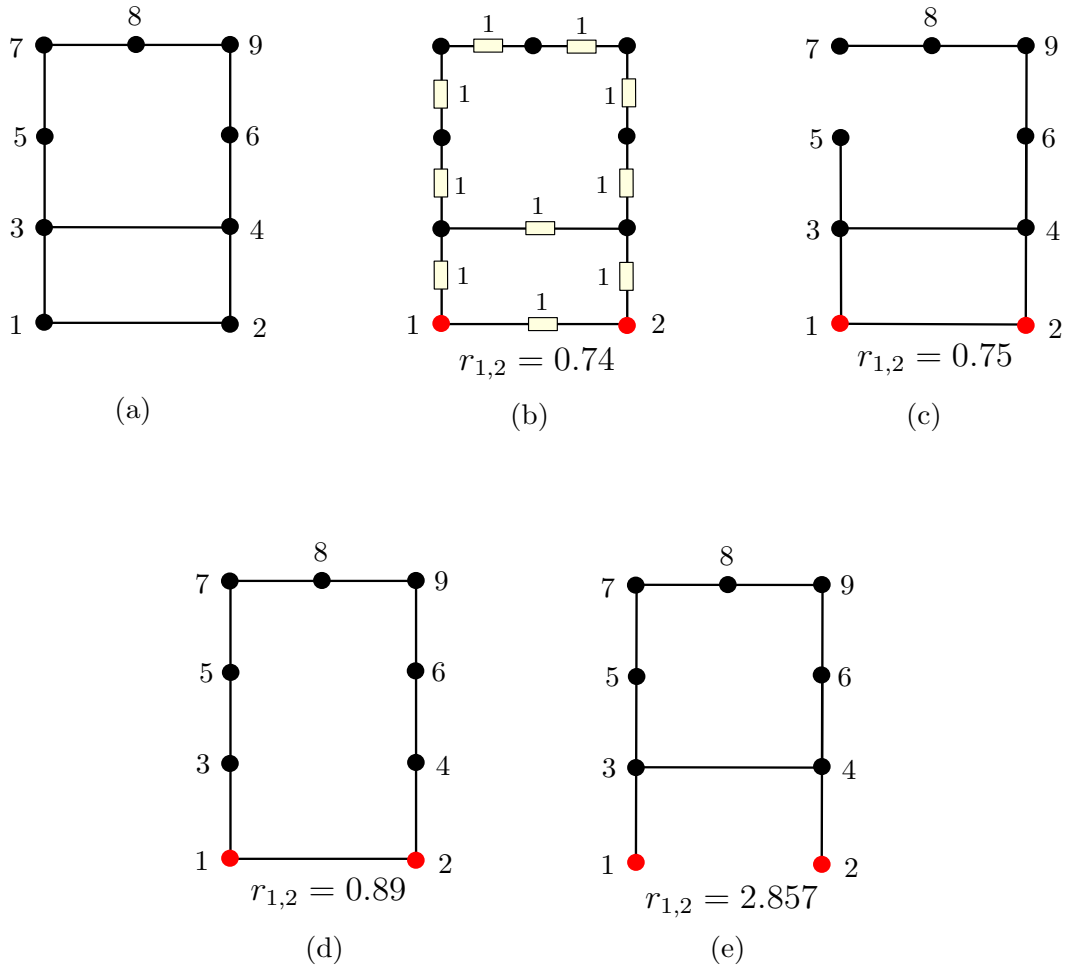


Figure 6.1: (a) A graph with eight nodes. (b) Each edge is replaced by a unit resistance and effective resistance between nodes 1 and 2 is calculated. In (c), (d) and (e), various edges are lost resulting in a loss of path between nodes 1 and 2. A corresponding increase in $r_{1,2}$ is also shown. Note that a smaller $r_{1,2}$ indicates a robust inter-connection between nodes 1 and 2.

Thus, the structural robustness of the overall network can be measured by the sum of the effective resistances over all pairs of nodes in the underlying graph, which is the so called *Kirchhoff index*, K_f, of the graph.

$$K_f(G) = \sum_{i=1}^{n} \sum_{j>i}^{n} r_{i,j} \tag{43}$$

where n is the number of vertices in G, and $r_{i,j}$ is an effective resistance between

nodes i and j.

A smaller value of K_f, indicates that a network is structurally more robust. It is also interesting to see that addition of an edge strictly decreases the value of K_f in a graph (shown in [90]), thus, increasing robustness. This also supports our intuition as addition of an edge always results in an extra path between a pair of nodes.

For the case of network robustness against noisy measurements, i.e. *functional robustness*, we consider a multiagent system with agents implementing a linear consensus protocol. Linear consensus dynamics have been extensively studied in the domain of network control systems due to its wide variety of applications including formation control, distributed control mechanisms, sensor networks and cooperative decision making to name a few (see [114]). Simple consensus dynamics of such a system can be given as,

$$\dot{x}(t) = -Lx(t) \tag{44}$$

where L is a laplacian matrix of an underlying graph and x is a corresponding state vector of the agents. In steady state, agents reach an agreement over a common state $\bar{x}(t)$. But for practical systems, agents' states are affected by a noise term. Thus,

$$\dot{x}(t) = -Lx(t) + \xi(t) \tag{45}$$

$\xi(t)$ is zero-mean mutually white stochastic process. It is known (e.g., see [91] and [95]) that in the presence of this noise term, agents' states do not converge at a common value but will remain in motion about $\bar{x}(t)$. In [91], robustness of a system in (45) under noisy consensus dynamics is then defined in terms of the expected dispersion of the system from consensus. A nice result reported there relates this robustness due to noisy consensus under the above settings to the Kirchhoff index of the undirected graph structure of the underlying network. It is shown that a network with a greater Kirchhoff index has a greater dispersion from consensus due to noise and is therefore, less robust. Similarly, a smaller value of K_f indicates that the expected dispersion of

the system in (45) due to noise is not significant, thus, indicating a greater robustness of network against noise.

In the light of the above discussion, it can be stated that seemingly different notions of structural robustness and functional robustness, are in fact, very inter-related. Both of them depend on the structure of an underlying network and can be measured by a same graph invariant known as the Kirchhoff index.

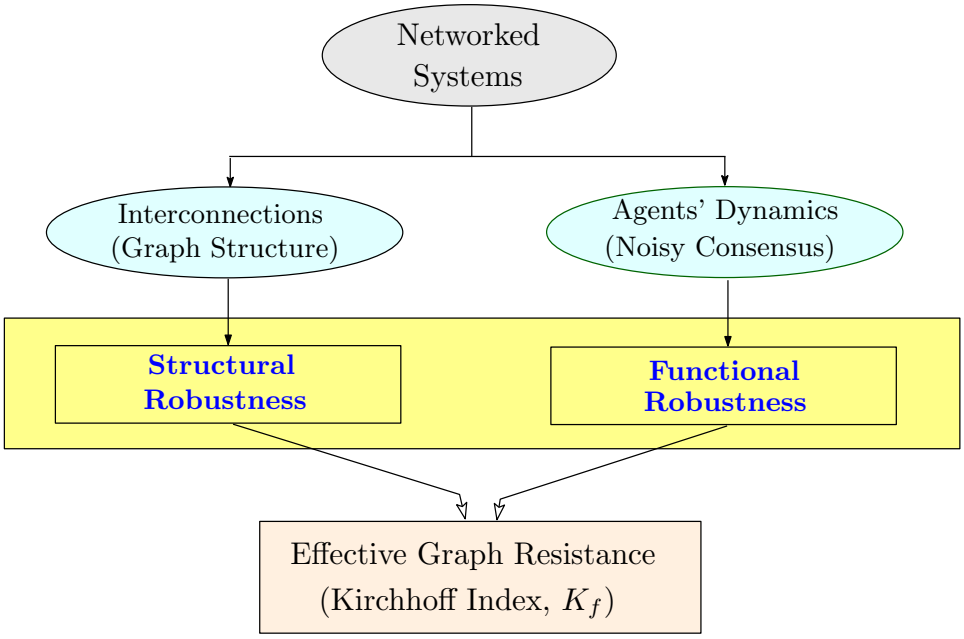


Figure 6.2: Structural robustness and functional robustness (robustness against noise) can both be quantified by Kirchhoff index.

6.2 Kirchhoff Indices of Some Graphs

As discussed, Kirchhoff index can measure both structural and functional robustness in multiagent systems. This provides us a way to develop a systematic scheme for designing optimal network topologies to maximize their robustness properties. In this section, we find Kirchhoff indices of various graph structures and also present optimal[1] addition of edges for some specific graphs. These results will be used in Section 6.3, in which a graph process is introduced in which edges are added to an

[1] in the sense of minimizing the Kirchhoff index, K_f.

existing graph using greedy approach to minimize the Kirchhoff index. At first, some graph terminologies are introduced.

A *Star Graph*, $\mathcal{S}_m$, is a tree with m vertices where $m-1$ vertices have a degree 1 and they all are connected to a single central vertex that has a degree $m-1$. A *Fan Graph*, $\mathcal{F}_m$ is obtained by connecting all the vertices in a path graph, $\mathcal{P}_{m+1}$, to a single vertex as shown in the Fig. 6.3.

Let G_1 and G_2 be two graphs, then $G_1 \bullet G_2$ denotes a graph obtained by identifying $u \in G_1$ with a vertex $v \in G_2$. An example is shown in Fig. 6.3. Moreover, if $v \in G_1$, then $(G_1)^k$ denotes a graph obtained by identifying k copies of G_1 through the vertex v, e.g., $(G_1)^3 = G_1 \bullet (G_1 \bullet G_1)$.

We also refer to $\mathcal{F}_i$ as an *i-petal*, and $(\mathcal{F}_i)^k$ as a *petal graph* containing k number of i-petals. An example is illustrated in the Fig. 6.3.

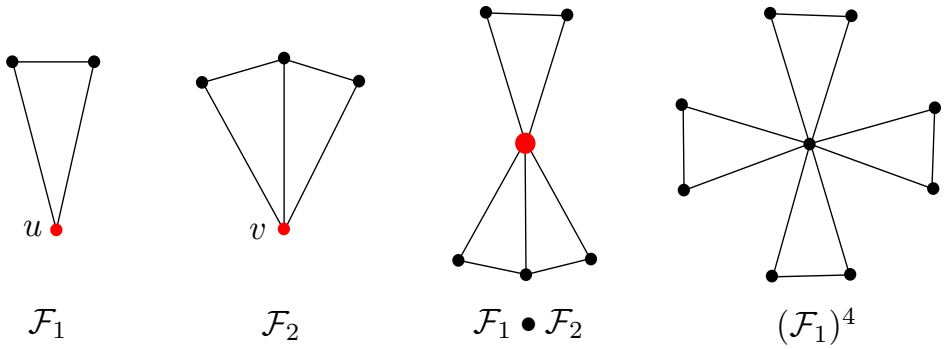


Figure 6.3: Fan graphs $\mathcal{F}_1$ and $\mathcal{F}_2$. Note that $\mathcal{F}_2$ is obtained by connecting all the vertices of a path graph with three nodes, $\mathcal{P}_3$, to a common node v. $\mathcal{F}_1 \bullet \mathcal{F}_2$ is obtained by identifying u and v vertices in $\mathcal{F}_1$ and $\mathcal{F}_2$ respectively. A petal graph, $(\mathcal{F}_1)^4$, with four 1-petals is also shown.

Lemma 6.2.1 *The Kirchhoff index of* $G = (\mathcal{F}_1)^k$ *is*

$$K_f((\mathcal{F}_1)^k) = \frac{2}{3}k(4k-1) \tag{46}$$

Proof. There are $2k+1$ vertices in $(\mathcal{F}_1)^k$. We label its vertices as $\{1, 2, \cdots, 2k, \alpha\}$, where α is the central vertex with a maximum degree as shown in the Fig. 6.4. Note

that if i is odd, $r_{i,i+1} = 2/3$ and $r_{i,j} = 4/3$ for every $j > i+1$. For even i, $r_{i,j} = 4/3$ for every $j > i$. Thus, for a fixed i,

$$\sum_{i<j} r_{i,j} = \begin{cases} \frac{4}{3}(2k-i) & i \text{ is even} \\ \\ \frac{2}{3} + \frac{4}{3}(2k-i-1) & i \text{ is odd} \end{cases}$$

Also, for every $i \in \{1, 2, \cdots, 2k\}$, we have $r_{i,\alpha} = 2/3$. Thus, Kirchhoff index of $(\mathcal{F}_1)^k$ can be written as,

$$K_f((\mathcal{F}_1)^k) = \sum_i r_{i,\alpha} + \sum_{i,j>i} r_{i,j}$$

After inserting the values and simplification we get,

$$K_f((\mathcal{F}_1)^k) = \frac{2}{3}(2k) + \left[\frac{8}{3}k(k-1) + \frac{2}{3}k\right] = \frac{2}{3}k(4k-1) \tag{47}$$

■

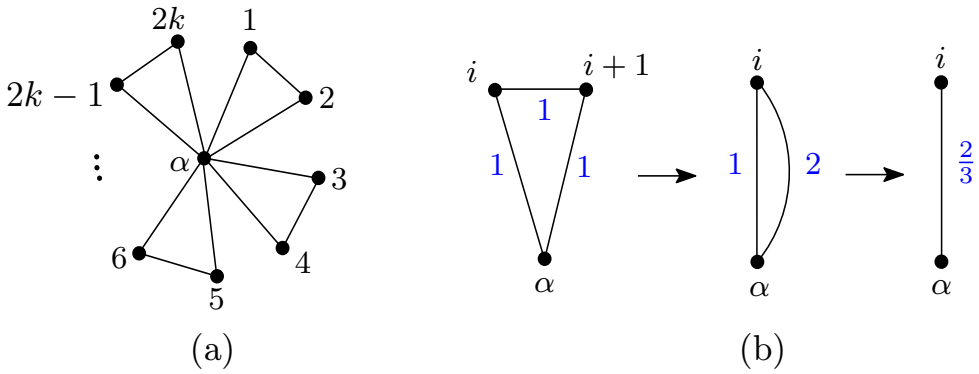


Figure 6.4: (a) Labeling of $(\mathcal{F}_1)^k$. (b) $r_{i,\alpha} = 2/3$.

A graph structure of the form $(\mathcal{F}_1)^k \bullet \mathcal{S}_m$, obtained by identifying a petal graph $(\mathcal{F}_1)^k$, and a star graph S_m through their central vertices, is used in the Section 6.3 for defining a graph process where edges are added to maximize robustness. Following lemma computes the K_f for such a graph.

Lemma 6.2.2 *Let $G = (\mathcal{F}_1)^k \bullet \mathcal{S}_m$ be a graph with $2k+m$ vertices. Then,*

$$K_f(G) = (m-1)^2 + \frac{2}{3}k(5m+4k-6) \tag{48}$$

Proof. Kirchhoff index of a given G can be written as,

$$K_f(G) = K_f((\mathcal{F}_1)^k) + K_f(\mathcal{S}_m) + \sum_{i\in\mathcal{S}_m, j\in(\mathcal{F}_1)^k} r_{i,j} \tag{49}$$

Let α be the central vertex of given G, (i.e., α is the vertex with a degree $2k+m$). Noting that $r_{i,\alpha} = 1$, where i is any of the non-central vertex of $\mathcal{S}_m$. Also, $r_{j,\alpha} = 2/3$, where j is any of the non-central vertex of $(\mathcal{F}_1)^k$. Thus, $r_{i,j} = 5/3$, where $i \in \mathcal{S}_m$ and $j \in (\mathcal{F}_1)^k$. This gives $\sum_{i\in\mathcal{S}_m, j\in(\mathcal{F}_1)^k} r_{i,j} = (m-1)\left[\frac{5}{3}(2k)\right]$. Also, we know that $K_f(\mathcal{S}_m) = (m-1)^2$ (see [90] as an example). Using these results along with (46), we get,

$$\begin{aligned} K_f(G) &= \frac{2}{3}k(4k-1) \ + \ (m-1)^2 \ + \ (m-1)\left[\frac{5}{3}(2k)\right] \\ &= (m-1)^2 + \frac{2}{3}k(5m+4k-6) \end{aligned}$$

■

We have also computed K_f for the following special graph structure that will be used later.

Lemma 6.2.3 *Let $G = (\mathcal{F}_1)^k \bullet \mathcal{S}_m$. Then the Kirchhoff index of $G' = G \bullet \mathcal{F}_2$ is*

$$K_f(G') = (m-1)^2 + \frac{19}{4}m - \frac{3}{4} + \frac{2k}{3}\left(5m + 4k + \frac{21}{4}\right) \tag{50}$$

Proof. Let us label the vertices in G' as shown in the Fig. 6.5. Here, α is the central vertex. The Kirchhoff index of G' can be written as,

$$K_f(G') = \ K_f(G) + K_f(\mathcal{F}_2) + \sum_{i\in\mathcal{F}_2, j\in G} r_{i,j}$$

Firstly we will find $\sum_{i\in\mathcal{F}_2, j\in G} r_{i,j}$. Since, $r_{a,\alpha} = r_{c,\alpha}$, so, $r_{a,j} = r_{c,j}$, where j is some vertex in G. Thus, we can write

$$\begin{aligned} \sum_{i\in\mathcal{F}_2, j\in G} r_{i,j} =& \sum_{j\in G} r_{a,j} \ + \ \sum_{j\in G} r_{b,j} \ + \ \sum_{j\in G} r_{c,j} \\ =& 2\sum_{j\in G} r_{a,j} \ + \ \sum_{j\in G} r_{b,j} \end{aligned} \tag{51}$$

Note that $r_{a,j} = r_{a,\alpha} + r_{\alpha,j}$, where $j \in G$, and $r_{a,\alpha} = 5/8$. Also, if j has a degree 1, then $r_{\alpha,j} = 1$ and if j has degree 2, $r_{\alpha,j} = 2/3$. This gives,

$$r_{a,j} = \begin{cases} 13/8 & j \in G \text{ and degree of } j = 1 \\ 31/24 & j \in G \text{ and degree of } j = 2 \end{cases}$$

Similarly, we have $r_{b,\alpha} = 1/2$. Thus, we get,

$$r_{b,j} = \begin{cases} 3/2 & j \in G \text{ and degree of } j = 1 \\ 7/6 & j \in G \text{ and degree of } j = 2 \end{cases}$$

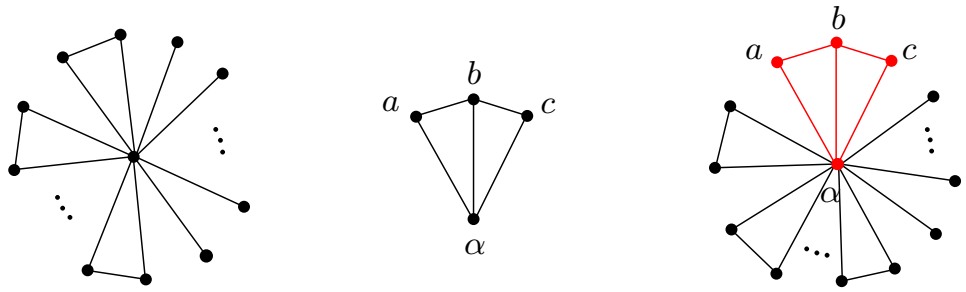
Since, G has $(m-1)$ vertices with degree 1 and $2k$ vertices with degree 2, thus inserting the $r_{a,j}$ and $r_{b,j}$ in (51), we get

$$\sum_{i \in \mathcal{F}_3, j \in G} r_{i,j} = \frac{19}{4}m + \frac{15}{2}k - \frac{19}{4} \tag{52}$$

Now, using (48), (52) and $K_f(\mathcal{F}_2) = 4$, we get,

$$K_f(G \bullet \mathcal{F}_2) = (m-1)^2 + \frac{19}{4}m - \frac{3}{4} + \frac{2k}{3}\left(5m + 4k + \frac{21}{4}\right)$$

■



$G = (\mathcal{F}_1)^k \bullet \mathcal{S}_m$ $\quad$ $G = \mathcal{F}_2$ $\quad$ $G' = G \bullet \mathcal{F}_2$

Figure 6.5: Labeling of $G' = G \bullet \mathcal{F}_2$, where $G = (\mathcal{F}_1)^k \bullet \mathcal{S}_m$. It is to be noted here that in case of $\mathcal{F}_2$, $r_{a,\alpha} = r_{c,\alpha} = r_{a,b} = r_{b,c} = 5/8$, $r_{b,\alpha} = 1/2$, and $r_{a,c} = 1$. Summing them all gives, $K_f(\mathcal{F}_2) = 4$.

Using exactly the same procedure in Lemma 6.2.3, we can prove the following result.

Lemma 6.2.4 *Let $G = (\mathcal{F}_1)^k \bullet \mathcal{S}_m$. Then the Kirchhoff index of $G' = G \bullet \mathcal{F}_3$ is*

$$K_f(G') = (m-1)^2 + \frac{130}{21}m + \frac{16}{21} + \frac{2k}{3}\left(5m + 4k + \frac{60}{7}\right) \tag{53}$$

Proof. Let us label the vertices in G' as shown in the Fig. 6.6. Here, α is the central vertex. The Kirchhoff index of G' can be written as,

$$K_f(G') = \ K_f(G) + K_f(\mathcal{F}_3) + \sum_{i\in\mathcal{F}_3, j\in G} r_{i,j}$$

Firstly we will find $\sum_{i\in\mathcal{F}_3, j\in G} r_{i,j}$. Note that,

$$\begin{aligned}\sum_{i\in\mathcal{F}_3, j\in G} r_{i,j} =& \sum_{j\in G} r_{a,j} \ + \ \sum_{j\in G} r_{b,j} \ + \ \sum_{j\in G} r_{c,j} \ + \ \sum_{j\in G} r_{d,j} \\ =& 2\sum_{j\in G} r_{a,j} \ + \ 2\sum_{j\in G} r_{b,j}\end{aligned} \tag{54}$$

since, $r_{a,j} = r_{d,j}$ and $r_{b,j} = r_{c,j}$. Note that $r_{a,j} = r_{a,\alpha} + r_{\alpha,j}$, where $j \in G$, and $r_{a,\alpha} = 13/21$. Also, if j has a degree 1, then $r_{\alpha,j} = 1$ and if j has degree 2, $r_{\alpha,j} = 2/3$. This gives,

$$r_{a,j} = \begin{cases} 34/21 & j \in G \text{ and degree of } j = 1 \\ 24/21 & j \in G \text{ and degree of } j = 2 \end{cases}$$

Similarly, we have $r_{b,\alpha} = 10/21$. Thus, we get,

$$r_{b,j} = \begin{cases} 31/21 & j \in G \text{ and degree of } j = 1 \\ 27/21 & j \in G \text{ and degree of } j = 2 \end{cases}$$

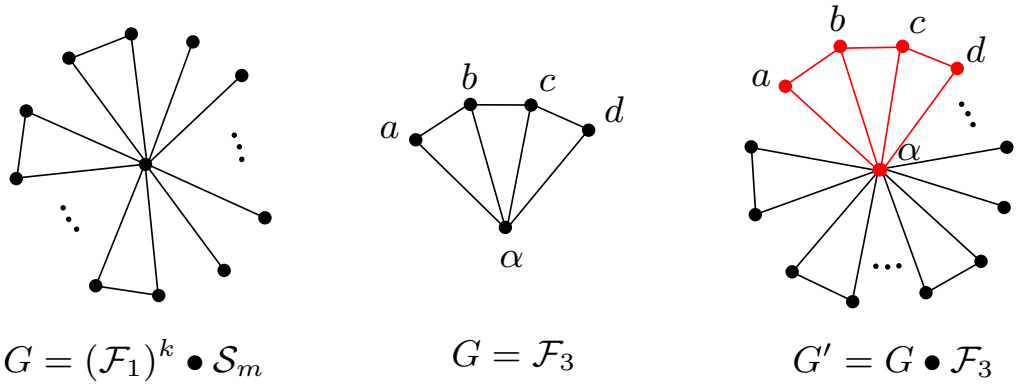
Since, G has $(m-1)$ vertices with degree 1 and $2k$ vertices with degree 2, thus inserting the $r_{a,j}$ and $r_{b,j}$ in (54), we get

$$\sum_{i\in\mathcal{F}_3, j\in G} r_{i,j} = \frac{130}{21}m + \frac{204}{21}k - \frac{130}{21} \tag{55}$$

Now, using (48), (55) and $K_f(\mathcal{F}_3) = 146/21$, we get,

$$K_f(G \bullet \mathcal{F}_3) = (m-1)^2 + \frac{130}{21}m + \frac{16}{21} + \frac{2k}{3}\left(5m + 4k + \frac{60}{7}\right)$$

■



$G = (\mathcal{F}_1)^k \bullet \mathcal{S}_m$ $\quad$ $G = \mathcal{F}_3$ $\quad$ $G' = G \bullet \mathcal{F}_3$

Figure 6.6: Labeling of $G' = G \bullet \mathcal{F}_3$, where $G = (\mathcal{F}_1)^k \bullet \mathcal{S}_m$. Here, $K_f(\mathcal{F}_3) = 146/21$.

Using these results, we can figure out the best way to add an edge in a graph $G = (\mathcal{F}_1)^k \bullet \mathcal{S}_m$, that will be required to optimally add edges in a graph in a step-wise manner.

Theorem 6.2.1 *Let $G = (\mathcal{F}_1)^k \bullet \mathcal{S}_m$ where $m > 1$, and H be a graph obtained from G by adding a single edge. Among all such H, $(\mathcal{F}_1)^{k+1} \bullet \mathcal{S}_{m-2}$ has a minimum value of Kirchhoff index.*

Proof. Let H be a graph obtained by adding an edge $u \sim v$ between any two non-adjacent vertices in $G = (\mathcal{F}_1)^k \bullet \mathcal{S}_m$. Then H is isomorphic to one of the following graphs,

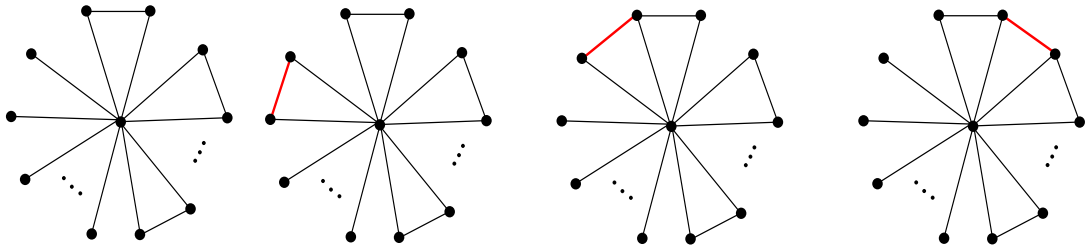
(1) $(\mathcal{F}_1)^{k+1} \bullet \mathcal{S}_{m-2}$

(2) $\mathcal{F}_2 \bullet ((\mathcal{F}_1)^{k-1} \bullet \mathcal{S}_{m-1})$

(3) $\mathcal{F}_3 \bullet ((\mathcal{F}_1)^{k-2} \bullet \mathcal{S}_m)$

This is true as there are only three ways of adding an edge in a given G. An edge can be added between u and v in G where u and v are of degree 1 as shown in the Fig. 6.7(b). This results in $H = (\mathcal{F}_1)^{k+1} \bullet \mathcal{S}_{m-2}$. When u has a degree 1 and v has

degree 2 (equivalently v has degree 1 and u has a degree 2) in an added edge $u \sim v$, we get $H = \mathcal{F}_2 \bullet ((\mathcal{F}_1)^{k-1} \bullet \mathcal{S}_{m-1})$. This is shown in the Fig. 6.7(c). When both the end vertices of an edge added to G are of degree 2, we get $H = \mathcal{F}_3 \bullet ((\mathcal{F}_1)^{k-2} \bullet \mathcal{S}_m)$, shown in the Fig. 6.7(d).



$(\mathcal{F}_1)^k \bullet \mathcal{S}_m$ $(\mathcal{F}_1)^{k+1} \bullet \mathcal{S}_{m-2}$ $\mathcal{F}_2 \bullet ((\mathcal{F}_1)^{k-1} \bullet \mathcal{S}_{m-1})$ $\mathcal{F}_3 \bullet ((\mathcal{F}_1)^{k-2} \bullet \mathcal{S}_m)$

(a) (b) (c) (d)

Figure 6.7: (a) $(\mathcal{F}_1)^k \bullet \mathcal{S}_m$. Adding an edge to (a) will result into one of the graphs shown in (b), (c) or (d).

Now let $H_1 = (\mathcal{F}_1)^{k+1} \bullet \mathcal{S}_{m-2}$, $H_2 = \mathcal{F}_2 \bullet ((\mathcal{F}_1)^{k-1} \bullet \mathcal{S}_{m-1})$ and $H_3 = \mathcal{F}_3 \bullet ((\mathcal{F}_1)^{k-2} \bullet \mathcal{S}_m)$. Each of these H_1, H_2 and H_3 have same number of edges and are obtained by adding a single edge in G.

Now using (48) and (53), we calculate $K_f(H_3) - K_f(H_1)$ as,

$$K_f(H_3) - K_f(H_1) = \frac{4}{21}(2k + m) > 0 \tag{56}$$

Similary using (48) and (50),

$$K_f(H_2) - K_f(H_1) = \frac{1}{12}(2k + m) > 0 \tag{57}$$

From (56) and (57), we have the following order

$$K_f(H_1) < K_f(H_2) < K_f(H_3)$$

which proves the desired result. ∎

Similarly, we can determine the best way to add an edge in a graph $(\mathcal{F}_1)^k \bullet (\mathcal{F}_3)^\ell$ to minimize the Kirchhoff index. This is crucial for defining a graph process to optimally

add edges in a step-wise manner as discussed in the Section 6.3. The proof of the following theorem is given in the Appendix B.

Theorem 6.2.2 *Let $k > 1$ and $G = (\mathcal{F}_1)^k \bullet (\mathcal{F}_3)^\ell$. Let H be a graph obtained from G by adding a single edge. Among all such H, $(\mathcal{F}_1)^{k-2} \bullet (\mathcal{F}_3)^{\ell+1}$ has a minimum value of Kirchhoff index.*

Proof. See B.

6.3 Graph Process for Step-wise Optimal Addition of Edges

Addition of an edge in a graph always decreases its K_f (as shown in [90]) and hence, increases robustness. But, addition of a certain missing edge may result in a greater decrease in K_f as compared to another edge. Thus, analysis regarding optimal addition of edges to minimize the Kirchhoff index is of great significance. As it is discussed in [90], the question of determining an optimal edge to add to a graph in order to minimize its K_f is still open. In this section, we provide a systematic way to obtain robust network topologies by optimally adding edges to existing graph structures. We start with a set of nodes without any edge between them, and successively add edges (one at at time), to maximally increase robustness. A notion of Kirchhoff graph process is introduced to characterize such a scheme.

Definition 6.3.1 *(Kirchhoff Graph Process): A Kirchhoff graph process, $\mathcal{G}$, on n vertices is a sequence of graphs, in which $\mathcal{G}_1$ is an edgeless graph on n vertices, and $\mathcal{G}_{i+1}$ is obtained by adding a single edge to $\mathcal{G}_i$ such that $\mathcal{G}_{i+1}$ has a minimum value of Kirchhoff index over all possible choices of $(\mathcal{G}_i + e)$, where $(\mathcal{G}_i + e)$ is a graph obtained by adding a single edge to $\mathcal{G}_i$.*

6.3.1 Kirchhoff Graph Process from $\mathcal{G}_1$ to $\mathcal{G}_n$

Note that the number of edges in $\mathcal{G}_i$ is $i - 1$. Since there are n nodes, so the graph will remain disconnected till $n - 1$ step. We know that a graph with n nodes and

$n-1$ edges with a minimum K_f is a star graph, $\mathcal{S}_n$ (e.g., see [92]). So, from $i=1$ to $i=n$, edges will be added so as to get $\mathcal{G}_n = \mathcal{S}_n$. Thus,

$$\mathcal{G}_i = \mathcal{S}_i \cup \bar{K}_{n-i} \qquad\qquad i \in \{1, 2, \cdots, n\} \tag{58}$$

where, $\bar{K}_{n-i}$ is an edgeless graph with $n-i$ nodes.

6.3.2 Kirchhoff Graph Process from $\mathcal{G}_{n+1}$ to $\mathcal{G}_{n+\lfloor\frac{n-1}{2}\rfloor}$

Adding an edge to a star graph, $\mathcal{S}_n$ always results in a $\mathcal{F}_1 \bullet \mathcal{S}_{n-2}$ graph. Thus, $\mathcal{G}_{n+1} = \mathcal{F}_1 \bullet \mathcal{S}_{n-2}$. The optimal way to add an edge in subsequent steps is to connect two non-adjacent vertices having a degree 1 as shown in Fig. 6.8(b). In fact, Theorem 6.2.1 and Lemma 6.2.2 provides the optimal way to add an edge in $(\mathcal{F}_1^k) \bullet \mathcal{S}_m$. Using these results, we get instances of the Kirchhoff graph process $\mathcal{G}_i$ for $i \in \{n+1, \cdots, \lfloor\frac{n-1}{2}\rfloor\}$ as,

$$\mathcal{G}_{n+i} = (\mathcal{F}_1)^i \bullet \mathcal{S}_{n-2i} \qquad i \in \{1, 2, \cdots, \lfloor\frac{n-1}{2}\rfloor\} \tag{59}$$

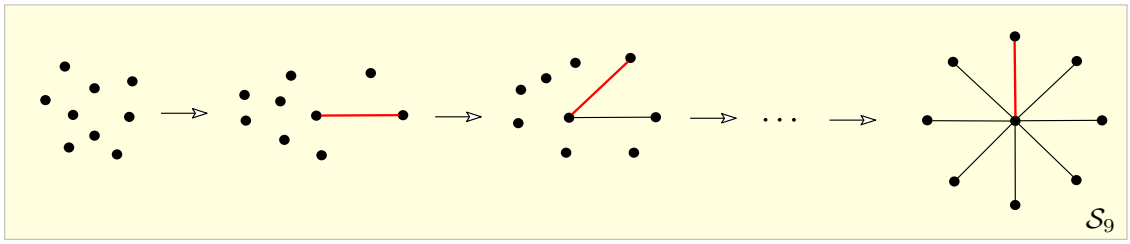
For a simpler case, let n be an odd number. Then, for $i = \left(\frac{n-1}{2}\right)$, $\mathcal{G}_{n+i}$ is a petal graph, $(\mathcal{F}_1)^{\frac{n-1}{2}} \bullet \mathcal{S}_1 = (\mathcal{F}_1)^{\frac{n-1}{2}}$.

6.3.3 Adding edges to a Petal Graph

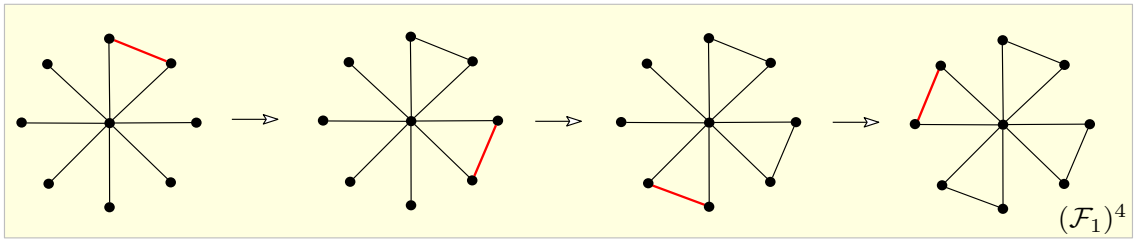
Adding an edge to a petal graph of the form $(\mathcal{F}_1)^k$, always results in a graph $(\mathcal{F}_1)^{k-2} \bullet \mathcal{F}_3$. Thus, in a Kirchhoff graph process,

$$\mathcal{G}_i = (\mathcal{F}_1)^{(\frac{n-1}{2}-2)} \bullet \mathcal{F}_3 \qquad i = n + \frac{n-1}{2} + 1 \tag{60}$$

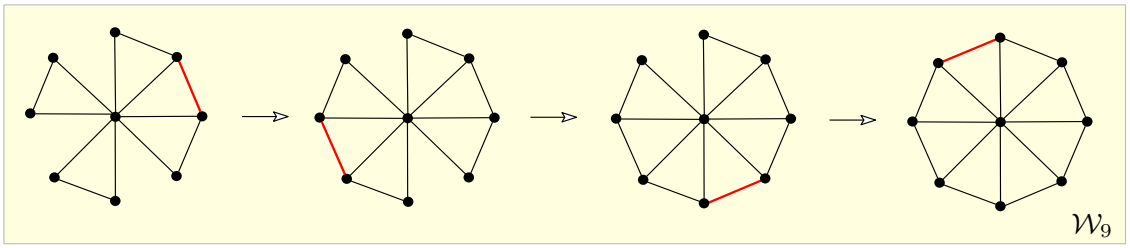
Moreover, as shown in Theorem 6.2.2, if a graph is of the form $(\mathcal{F}_1)^k \bullet (\mathcal{F}_3)^\ell$, then the optimal addition of an edge minimizing the Kirchhoff index yields a graph $(\mathcal{F}_1)^{k-2} \bullet (\mathcal{F}_3)^{\ell+1}$. This results provides a way of adding edges to instances of a graph process $\mathcal{G}_i$ for $i > n + \left(\frac{n-1}{2}\right)$. An example is also shown in the Fig. 6.8(c). Further analysis of this process shows that edges are being added in a specific pattern. From a star graph at $\mathcal{G}_n = \mathcal{S}_n$, edges are added to increase the number of 1-petals (i.e., $\mathcal{F}_1$) in



(a)



(b)



(c)

Figure 6.8: A Kirchhoff graph process for $n = 9$ nodes.

the intermediate steps of the Kirchhoff graph process until a petal graph is obtained, in which every petal is a 1-petal. Similarly, from a 1-petal graph at $\mathcal{G}_{n+\frac{n-1}{2}} = (\mathcal{F}_1)^{\frac{n-1}{2}}$, edges are added to increase the number of 3-petals (i.e. $\mathcal{F}_3$) by connecting two 1-petals. This continues till a petal graph is obtained, in which every petal is a 3-petal. In the next steps, edges are added to 3-petal graph such that at each step two 3-petals are combined to get a 7-petal. This continues until a wheel graph $\mathcal{W}_n$ is obtained at the $2n-1$ step of the Kirchhoff graph process, that is,

$$\mathcal{G}_{2n-1} = \mathcal{W}_n \tag{61}$$

It is to be noted here that at each step of the Kirchhoff graph process, an edge is added optimally to maximize the robustness property of the graph. An example for $n = 9$ vertices is shown in the Fig. 6.8.

6.3.4 Step-wise Optimal Graph vs. Globally Optimal Graph

Consider a graph with n vertices for some odd integer n, containing $(n-1)+\left(\frac{n-1}{2}\right)$ edges, and obtained through a Kirchhoff graph process. From (59), we know that it is a graph of the form $(\mathcal{F}_1)^{\frac{n-1}{2}}$. Its Kirchhoff index can be computed using Lemma 6.2.1 for any n. A gear graph[2] with n verices and $(\mathcal{F}_1)^{\frac{n-1}{2}}$ has same number of vertices and edges. It is observed that for a number of values of n, a gear graph with n vertices has a smaller K_f than $(\mathcal{F}_1)^{\frac{n-1}{2}}$. This implies that although $(\mathcal{F}_1)^{\frac{n-1}{2}}$ is obtained by optimally adding edges in a step-wise manner, still it is not a graph with a minimum K_f for a given number of nodes and edges.

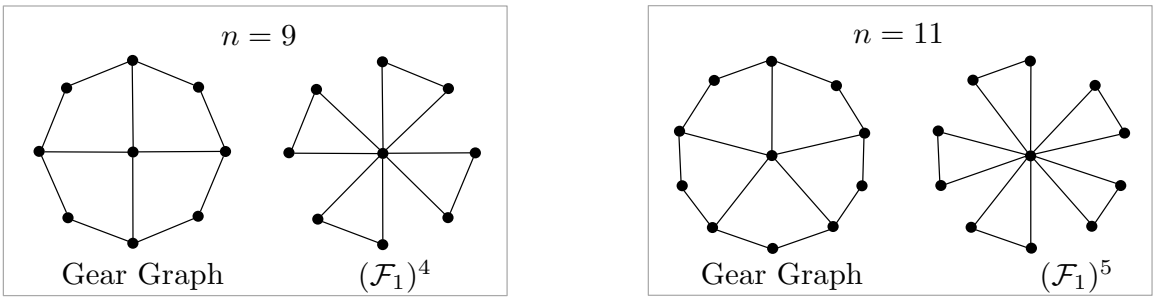


Figure 6.9: (a) A gear graph with 9 nodes and a petal graph, $(\mathcal{F}_1)^4$. (b) A gear graph with 11 nodes and $(\mathcal{F}_1)^5$. In both cases, gear graph has a smaller K_f.

A comparison of K_f values for a gear graph and a petal graph with the same number of nodes and edges is shown in the Table 6.3.4. Thus, optimal step-wise addition of edges does not necessarily give a globally optimum graph, i.e. a graph with a minimum K_f for a given number of nodes and edges. We can state it as a following Proposition.

Proposition 7 *A graph G with $\mathcal{E}$ number of edges, obtained through a Kirchhoff graph process by optimally adding a single edge at each step of the process to minimize*

[2]A gear graph with $2m+1$ vertices is obtained from a wheel graph $\mathcal{W}_m$, by adding a vertex between each pair of adjacent vertices on the outer cycle of $\mathcal{W}_m$ (see Fig. 6.9).

K_f, does not necessarily give a globally optimum graph with the minimum K_f among all graphs with n nodes and $\mathcal{E}$ edges.

Table 2: Comparison of K_f of gear graph and $(\mathcal{F}_1)^k$ with the same number of vertices, and edges.

No. of Vertices	K_f(Gear graph)	K_f $((\mathcal{F}_1)^{\frac{n-1}{2}})$
9	34.5	40
11	57.11	63.33
13	85.67	92
15	120.08	136
17	160.31	165.33
19	206.32	210

6.3.5 Effect of Vertex Removal on the Kirchhoff Index of a Graph

It is known that removal of an edge strictly increases the Kirchhoff index of a graph, thereby making it less robust (see [90]). It is interesting to see the effect of a vertex removal on the robustness properties of the resulting graph. Unlike edge removal, a vertex removal does not always increase K_f, but may also result in the decrease of K_f. This is demonstrated in the two examples shown in the Fig. 6.10 and Fig. 6.11.

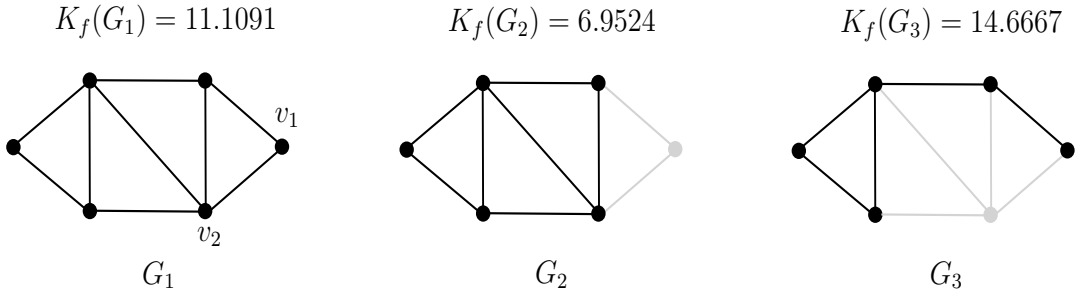


Figure 6.10: Deletion of vertex v_1 from the graph G_1 results in the graph G_2. Note that G_2 has a smaller K_f. On the other hand, removing vertex v_2 from G_1 results into G_3 that has a higher K_f than G_1, and therefore less robust than G_1.

It can be shown that removing a vertex of degree 1 from a graph results in a graph that has a smaller K_f compared to the original graph. A very useful definition of

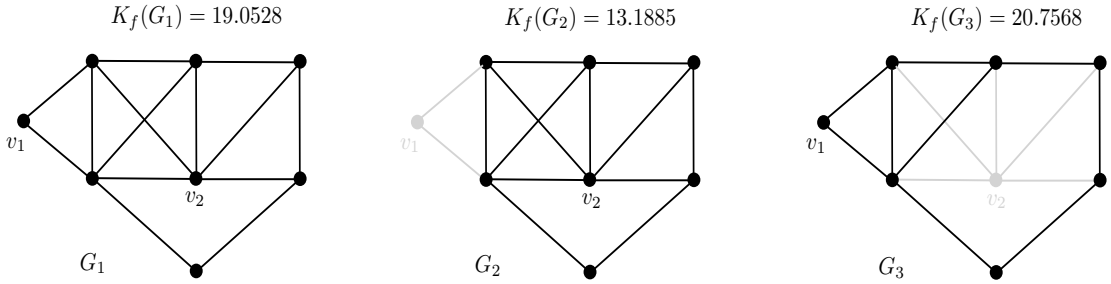


Figure 6.11: Removing v_1 from G_1 results into a more robust graph G_2 while removal of a vertex v_2 results into G_3 that has a higher K_f than G_1.

$K_f(G)$ has been provided in [94], in which K_f is defined in terms of the eigen values of the Laplacian matrix of a graph.

Theorem 6.3.1 *([94]) Let $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_{n-1} > \lambda_n = 0$ be the eigen values of the Laplacian matrix of a graph G with n vertices, then,*

$$K_f(G) = n \left(\sum_{i=1}^{n-1} \frac{1}{\lambda_i} \right) \tag{62}$$

Another useful result relating the Laplacian eigen values of G with the eigen values of the Laplacian matrix of $(G - v)$, where $(G - v)$ is a graph obtained by deleting a vertex v from G, is provided in [122].

Theorem 6.3.2 *([122]) Let G be a graph of order n and $H = G - v$, where v is a vertex of G of degree r. If $\lambda_1 \geq \lambda_2 \cdots \geq \lambda_n = 0$ and $\theta_1 \geq \theta_2 \cdots \geq \theta_{n-1} = 0$, are the eigen values of $L(G)$ and $L(H)$ respectively, then,*

$$\lambda_i \geq \theta_i \geq \lambda_{i+r} \tag{63}$$

for each $i = 1, 2, \cdots, n-1$, where $\lambda_i = 0$ for $i \geq n+1$.

Using (62) and (63), it can be shown that,

Lemma 6.3.1 *Let v be a vertex of degree 1 in G, then,*

$$K_f(G - v) \; < \; K_f(G) \tag{64}$$

Proof. Let G be a graph of order n and $H = G - v$, where v is a vertex of G of degree 1. Also $\lambda_1 \geq \lambda_2 \cdots \geq \lambda_n = 0$ and $\theta_1 \geq \theta_2 \cdots \geq \theta_{n-1} = 0$, be the eigen values of $L(G)$ and $L(H)$ respectively, then, using (63),

$$\lambda_1 \geq \theta_1 \geq \lambda_2 \geq \theta_2 \cdots \geq \lambda_{n-2} \geq \theta_{n-2} \geq \lambda_{n-1}$$

or,

$$\frac{1}{\lambda_1} \leq \frac{1}{\theta_1} \leq \frac{1}{\lambda_2} \leq \frac{1}{\theta_2} \cdots \leq \frac{1}{\lambda_{n-2}} \leq \frac{1}{\theta_{n-2}} \leq \frac{1}{\lambda_{n-1}}$$

This gives us,

$$\sum_{i=1}^{n-2} \frac{1}{\theta_i} \;\leq\; \sum_{i=2}^{n-1} \frac{1}{\lambda_i} \;<\; \sum_{i=1}^{n-1} \frac{1}{\lambda_i}$$

Using (62), we get our desired result, $K_f(G-v) \;<\; K_f(G)$. ∎

Similarly, using (62) and (63), we can provide a sufficient condition for $K_f(G - v) \;<\; K_f(G)$, where v is a vertex of degree r.

Lemma 6.3.2 *Let G be a graph with a vertex v of degree r. If,*

$$\sum_{i=1}^{r} \frac{1}{\lambda_i} \;>\; \sum_{i=n-r}^{n-2} \frac{1}{\theta_i} \tag{65}$$

then,

$$K_f(G-v) \;<\; K_f(G)$$

Proof. Using (63),

$$\sum_{i=1}^{n-r-1} 1/\theta_i \leq \sum_{i=r+1}^{n-1} 1/\lambda_i$$

Adding $\displaystyle\sum_{i=n-r}^{n-2} 1/\theta_i$ on both sides,

$$\sum_{i=1}^{n-2} 1/\theta_i \;\leq\; \sum_{i=r+1}^{n-1} 1/\lambda_i \;+\; \sum_{i=n-r}^{n-2} 1/\theta_i$$

Using (62), we get,

$$\frac{K_f(G-v)}{n-1} \leq \sum_{i=1}^{n-1} 1/\lambda_i - \sum_{i=1}^{r} 1/\lambda_i + \sum_{i=n-r}^{n-2} 1/\theta_i = \frac{K_f(G)}{n} - \left[\sum_{i=1}^{r} 1/\lambda_i - \sum_{i=n-r}^{n-2} 1/\theta_i\right]$$

Thus, we get,

$$\frac{K_f(G-v)}{n(n-1)} < \frac{1}{n(n-1)}\left(K_f(G) - n\left[\sum_{i=1}^{r} 1/\lambda_i - \sum_{i=n-r}^{n-2} 1/\theta_i\right]\right)$$

$$K_f(G-v) < K_f(G) - n\left[\sum_{i=1}^{r} 1/\lambda_i - \sum_{i=n-r}^{n-2} 1/\theta_i\right]$$

From here, if we have,

$$\sum_{i=1}^{r} 1/\lambda_i > \sum_{i=n-r}^{n-2} 1/\theta_i$$

then, $K_f(G-v) < K_f(G)$, which is the required result. ∎

An example demonstrating Lemma 6.3.1 is shown in the Fig. 6.12.

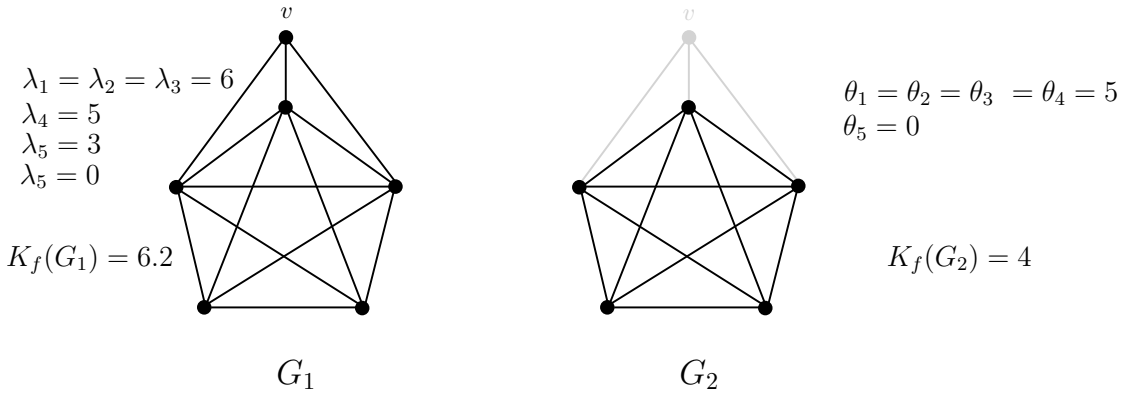


Figure 6.12: A graph G_1 has $K_f(G_1) = 6.2$. Deleting a vertex v having a degree 3 results into a new graph G_2. Since, $\sum_{i=1}^{3} 1/\lambda_i = 0.5$ and $\sum_{i=3}^{4} 1/\theta_i = 0.4$, so the condition in (65) is satisfied and so $K_f(G_2) < K_f(G_1)$, which is also verified by $K_f(G_2) = 4$.

6.4 Symmetry of Networks and Robustness

Symmetric network topologies are more robust and have a smaller Kirchhoff index (for example, see [90] and [97]). In fact, for a given number of nodes and diameter,

a special graph known as a *clique chain* (see [90]), which is a symmetric structure, has the minimum value of effective resistance and therefore, maximum robustness. Similarly for a given number of nodes, a complete graph which is also symmetric, has the maximum robustness. A relationship between symmetry and robustness can also be seen in the Kirchhoff graph process discussed in Section 6.3. At each step of the process, edges are added so as to preserve the symmetry of the overall graph. Thus, symmetry of a graph has a far reaching impact on its robustness properties.

Here, we show an optimal (in the sense of minimizing the K_f) way to attach a path graph to an arbitrary graph G. Again it is observed that symmetry of a graph plays an important role in minimizing K_f. Let G be any graph with j number of nodes, where $j > 1$. A *vine graph* is obtained from a graph G by attaching two separate paths with i and p number of nodes to G through two of its nodes. Let a path $\mathcal{P}_i$ be connected to G through node 1 and a path $\mathcal{P}_p$ through node j of G. A vine graph, denoted by $G_{\{i,p\}}$ is shown in the Fig. 6.13 (a). In a vine graph, paths of i and p nodes may be connected to G through the same vertex, as shown in the Fig. 6.13 (b). In [92], it is shown that if paths $\mathcal{P}_i$ and $\mathcal{P}_p$, where $1 \leq i \leq p$, are connected to a tree graph, denoted by T, through the same vertex, then,

$$K_f\left(T_{\{i,p\}}\right) < K_f\left(T_{\{i-1,p+1\}}\right) \tag{66}$$

Here, we generalize this result and show that (66) holds even if trees are replaced with any other graphs. In fact, we provide a necessary and sufficient condition for $K_f\left(G_{\{i,p\}}\right) < K_f\left(G_{\{i-1,p+1\}}\right)$ to be true even when paths with i and p number of nodes are connected to G through two different vertices, say 1 and j respectively.

Theorem 6.4.1 *Let G be a graph with $j > 1$ nodes. Let a path $\mathcal{P}_i$ be connected to G through a node, say 1, of G. Another path, $\mathcal{P}_p$ be connected to G through a node, say j, of G, to get a vine graph $G_{\{i,p\}}$, where, $1 \leq i \leq p$. Then*

$$K_f\left(G_{\{i,p\}}\right) < K_f\left(G_{\{i-1,p+1\}}\right) \tag{67}$$

whenever

$$(p+1-i)(j-1-r_{1,j}) > \sum_{s=1}^{j} r_{1,s} - \sum_{s=1}^{j} r_{s,j} \tag{68}$$

Proof. Without loss of generality, let us label the vertices in $G_{\{i,p\}}$ as shown in Fig. 6.13. Then, we can write the K_f of $G_{\{i,p\}}$ as follows,

$$\begin{aligned} K_f(G_{\{i,p\}}) &= \sum_{1\le s<t\le(j+i+p)} r_{s,t} \\ &= \sum_{s=1}^{j}\sum_{t>s}^{j} r_{s,t} \; + \underbrace{\sum_{s=1}^{j}\sum_{t=(j+1)}^{(j+i+p)} r_{s,t}}_{A} + \underbrace{\sum_{s=(j+1)}^{(j+i+p)}\sum_{t>s}^{(j+i+p)} r_{s,t}}_{B} \end{aligned} \tag{69}$$

Computing the A term in (69).

$$\begin{aligned} A =& \sum_{s=1}^{j}\sum_{t=(j+1)}^{(j+i+p)} r_{s,t} \\ =& \sum_{t=(j+1)}^{(j+i+p)} r_{1,t} \; + \sum_{t=(j+1)}^{(j+i+p)} r_{2,t} \; + \; \cdots \sum_{t=(j+1)}^{(j+i+p)} r_{j,t} \\ =& [(1+2+\cdots+i) + p(r_{1,j}) + (1+2+\cdots+p)] + [ir_{1,2} + (1+2+\cdots+i) + pr_{2,j} \\ & + (1+2+\cdots+p)] \; + \; \cdots \; + \; [ir_{1,j} + (1+2+\cdots+i) + (1+2+\cdots+p)] \end{aligned}$$

After simplification we get,

$$A = \frac{ij}{2}(1+i) \; + \; \frac{jp}{2}(1+p) \; + \; p\sum_{s=1}^{j} r_{s,j} \; + \; i\sum_{s=1}^{j} r_{1,s} \tag{70}$$

Now, to find the B term in (69), we proceed as follows,

$$\begin{aligned} B &= \sum_{s=(j+1)}^{(j+i+p)}\sum_{t>s}^{(j+i+p)} r_{s,t} \\ &= \sum_{s=(j+1)}^{(j+i)}\sum_{t>s}^{(j+i)} r_{s,t} \; + \sum_{s=(j+i+1)}^{(j+i+p)}\sum_{t>s}^{(j+i+p)} r_{s,t} \; + \sum_{s=(j+1)}^{(j+i)}\sum_{j+i+1}^{(j+i+p)} r_{s,t} \end{aligned} \tag{71}$$

Note that the first two terms in (71) correspond to the Kirchhoff index of paths with i and p number of nodes respectively. So, we can write (71) as,

$$B = K_f(\mathcal{P}_i) \ + \ K_f(\mathcal{P}_p) \ + \ \sum_{s=(j+1)}^{(j+i)} \sum_{j+i+1}^{(j+i+p)} r_{s,t} \tag{72}$$

Now, we evaluate the last term in (72) as,

$$\begin{aligned}\sum_{s=(j+1)}^{(j+i)} \sum_{j+i+1}^{(j+i+p)} r_{s,t} = & [p(i + r_{1,j}) + (1 + \cdots + p)] \ + \ [p((i-1) + r_{1,j}) + (1 + \cdots + p)] \ + \\ & \cdots \ + \ [p(1 + r_{1,j}) + (1 + \cdots + p)] \\ & = i\left(\frac{p}{2}(p+1)\right) \ + \ ipr_{1,j} \ + \ p(1 + \cdots + i) \\ & = \frac{ip}{2}\left[i + p + 2r_{1,j} + 2\right]\end{aligned} \tag{73}$$

Kirchhoff Index of a path graph with x number of nodes can be easily calculated as,

$$K_f(\mathcal{P}_x) = \frac{1}{6}(x-1)(x)(x+1) \tag{74}$$

Inserting (73) and (74) into (72) gives,

$$\begin{aligned}B &= \frac{i}{6}(i-1)(i+1) \ + \ \frac{p}{6}(p-1)(p+1) \ + \ \frac{ip}{2}(i + p + 2r_{1,j} + 2) \\ &= \frac{1}{6}\left[i(i^2-1) + p(p^2-1)\right] \ + \ \frac{ip}{2}(i + p + 2r_{1,j} + 2)\end{aligned} \tag{75}$$

Now inserting A from (70) and B from (75) in (69) gives $K_f(G_{\{i,p\}})$ as,

$$\begin{aligned}K_f(G_{\{i,p\}}) = & \sum_{s=1}^{j} \sum_{t>s}^{j} r_{s,t} \ + \ \frac{ij}{2}(1+i) \ + \ \frac{jp}{2}(1+p) \ + \ p\sum_{s=1}^{j} r_{s,j} \ + \ i\sum_{s=1}^{j} r_{1,s} \\ & + \frac{1}{6}\left[i(i^2-1) + p(p^2-1)\right] \ + \ \frac{ip}{2}(i + p + 2r_{1,j} + 2)\end{aligned} \tag{76}$$

As in (69), we can write $K_f(G_{\{i-1,p+1\}})$ as follows,

$$K_f(G_{\{i-1,p+1\}}) = \sum_{1\le s<t\le(j+i+p)} r_{s,t}$$
$$= \sum_{s=1}^{j}\sum_{t>s}^{j} r_{s,t} \;+\; \underbrace{\sum_{s=1}^{j}\sum_{t=(j+1)}^{(j+i+p)} r_{s,t}}_{A'} + \underbrace{\sum_{s=(j+1)}^{(j+i+p)}\sum_{t>s}^{(j+i+p)} r_{s,t}}_{B'} \tag{77}$$

Here, we need to calculate the corresponding A' and B' terms. Using exactly the same procedure as for A and B, we get A' and B' as follows,

$$A' \;=\; \frac{ij}{2}(i-1) \;+\; \frac{j}{2}(p+1)(p+2) + \;(p+1)\sum_{s=1}^{j} r_{s,j} \;+\; (i-1)\sum_{s=1}^{j} r_{1,s} \tag{78}$$

$$B' \;=\; \frac{1}{6}\left[i(i^2-3i+2)+p(p^2+3p+2)\right] \;+\; \frac{1}{2}(i-1)(p+1)\left(p+i+2+2r_{1,j}\right) \tag{79}$$

Inserting (78) and (79), in (77) gives $K_f(G_{\{i-1,p+1\}})$,

$$\begin{aligned} K_f(G_{\{i-1,p+1\}}) = &\sum_{s=1}^{j}\sum_{t>s}^{j} r_{s,t} \;+\; \frac{ij}{2}(i-1) \;+\; \frac{j}{2}(p+1)(p+2) + \;(p+1)\sum_{s=1}^{j} r_{s,j} \\ &+\; (i-1)\sum_{s=1}^{j} r_{1,s} \;+\; \frac{1}{6}\left[i(i^2-3i+2)+p(p^2+3p+2)\right] \\ &+\; \frac{1}{2}(i-1)(p+1)\left(p+i+2+2r_{1,j}\right) \end{aligned} \tag{80}$$

Now, subtracting (76) from (80), and simplifying gives us the following,

$$\begin{aligned} K_f(G_{\{i-1,p+1\}}) - K_f(G_{\{i,p\}}) = &\sum_{s=1}^{j} r_{s,j} - \sum_{s=1}^{j} r_{1,s} \\ &+ (p+1-i)(j-1-r_{1,j}) \end{aligned} \tag{81}$$

The required result directly follows from (81). ∎

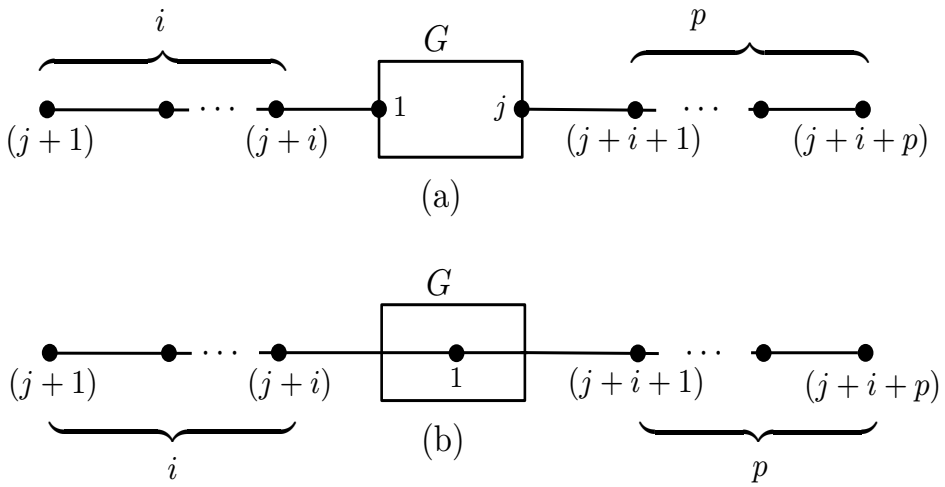


Figure 6.13: (a) Paths $\mathcal{P}_i$ and $\mathcal{P}_p$ are connected to G through vertices 1 and j respectively. In (b), both paths $\mathcal{P}_i$ and $\mathcal{P}_p$ are connected through the same vertex, 1.

A special case of the above theorem is when $\mathcal{P}_i$ and $\mathcal{P}_p$ are connected to G through the same vertex, say 1 (as shown in the Fig. 6.13(b)). The condition in (68) is always satisfied as long as $1 \leq i \leq p$. This is true as 1 and j in (68) correspond to the same vertex here and so, $\sum_{s=1}^{j} r_{1,s} = \sum_{s=1}^{j} r_{s,j}$. Thus, we get the following result,

Theorem 6.4.2 *Let G be a graph with at least two vertices. Let two paths with i and p number of vertices respectively, are connected to G through the same vertex of G to get $G_{\{i,p\}}$. Then,*

$$K_f(G_{\{i,p\}}) < K_f(G_{\{i-1,p+1\}}) \tag{82}$$

Here, $1 \leq i \leq p$.

Proof. We can use the condition (67) in Theorem 6.4.1. Here, paths with i and p number of vertices are connected to G through the same vertex, say 1 (see Fig. 6.13(b)). Then, vertices labeled 1 and j in the condition (67) are same, thus, $r_{1,x} = r_{j,x}$, for any vertex x. Thus, (67) becomes,

$$\begin{aligned} (p+1-i)(j-1-r_{1,j}) \; &> \; \sum_{s=1}^{j} r_{1,s} - \sum_{s=1}^{j} r_{s,j} \\ (p+1-i)(j-1) \; &> \; 0 \end{aligned} \tag{83}$$

It is to be noted that j here is the number of vertices in G, and under our assumption $j \geq 2$, thus (83) is always satisfied. Using Theorem 6.4.1, we get the required result directly. ∎

6.5 Discussion

It is to be mentioned here that the symmetry of underlying graph structure also plays an important role in determining some other properties of networked systems with agents implementing a linear consensus protocol. One such noticeable property is the controllability of such systems under a leader-follower setting, in which external inputs are injected through so called leader nodes. Structures that are symmetric about a leader exhibit poor controllability properties (see [114]). For example, a complete graph (most robust network for a given number of nodes) is least controllable. Thus, we can say that from a network topology perspective, controllability and robustness properties are in conflict with each other. Improving one by reconfiguring the underlying graph structure may deteriorate the other one. A precise relationship between these two properties in terms of the graph structure is an interesting research direction.

Several measures have been proposed to measure the robustness of network structure against link failures and edge removals. In all those propositions, the guiding principle is to design a metric that quantifies well-connectedness of the network at various levels. The notions of Kirchhoff index and effective resistances among nodes have been useful in this context in the sense that many of the other network robustness measures can be written in the form of Kirchhoff index and effective resistances. As an example, the number of spanning trees in the underlying graph of the network, denoted by $\tau(G)$, is sometimes used to measure the network robustness and ability to remain connected in the case of link failures [126]. $\tau(G)$ and the Kirchhoff index $K_f(G)$ are closely related to each other. In fact, if $x^n + c_{n-1}x^{n-1} + \cdots + c_2x^2 + c_1x$

is a characteristic polynomial of the Laplcaian matrix of G, then

$$K_f(G) \;=\; \frac{\mid c_2 \mid}{\tau(G)}$$

This is true because of the following: if $\lambda_1 \geq \lambda_2 \geq \cdots > \lambda_n = 0$ are the eigen values of the Laplacian matrix of G, then $\sum_{i=1}^{n-1} \frac{1}{\lambda_i} = \frac{|c_2|}{|c_1|}$ as shown in [123]. Using (62), $K_f(G) = n \sum_{i=1}^{n-1} \frac{1}{\lambda_i}$. Moreover, $\tau(G) = \frac{|c_1|}{n}$ by the matrix-tree theorem [123]. Thus, $K_f(G) = n\frac{|c_2|}{|c_1|} = \frac{|c_2|}{\tau(G)}$.

Similarly, other metrics that quantify significance of links on the overall connectivity and reliability of networks such as information centrality, betweenness centrality, and node vulnerability etc., can be written in terms of the Kirchhoff index and effective resistances (e.g., see [124, 125, 126]).

CHAPTER VII

CONCLUSIONS

This dissertation presented graph theoretic methods to analyze and explore the role of network topologies in networked systems when agents with a diverse set of resources and capabilities interact with each other to accomplish various tasks.

In Chapter 2, we studied heterogeneity in multiagent systems from a network topology view-point. The notion of (r, s)-configuration of a graph was used to characterize the distribution of agents with multiple capabilities (or resources). In such a distribution, every agent could find all types of resources available in the network in its closed neighborhood. The role of individual agents and interactions among them in attaining (r, s)-configurations was also examined. The study not only analysed the role of network topology in the context of heterogeneous multiagent systems, but also provided ways to design network structures, in which agents equipped with various resources coordinate and compliment each others capabilities to accomplish complex tasks.

The issue of energy-efficient data collection in heterogeneous wireless sensor and actor networks was addressed in Chapter 3. Using the framework introduced in Chapter 2, redundancy among sensors of various types, which were distributed at random with certain intensities, was explored. Sensors that were redundant in the sense that their deactivation did not affect the availability of data to the actors were determined and eventually turned off to save energy. Simulations performed showed that typically more than two-third of the sensors could be deactivated without compromising availability of data to the actors through our scheme.

Further, we explored the value of heterogeneity for the complete coverage problem for mobile agents with circular sensing areas in Chapter 4. It was shown that in terms of the coverage densities and sensing costs metrics, more efficient solutions can be obtained through networks of heterogeneous agents as compared to the homogeneous ones. In particular, configurations of agents with two different radii were examined that exhibited higher coverage densities and lower sensing costs than the configuration of agents having same radii. Moreover, we also analyzed coordination frameworks required to ensure that formations of agents in a particular configurations were maintained when agents exhibited movements.

In Chapters 5 and 6, role of network topologies for reliable functioning of multiagent systems was considered. In particular, the issue of securing a multiagent system against a sequence intruder attacks was investigated in Chapter 5 using constructs from graph theory. Special agents known as guards, where each guard could detect and respond to an intruder attack within some range, were distributed among vertices of a graph. Deployment of guards ensured that every vertex was situated within a range of at least one guard at all times. Moreover, coverage property was maintained and the graph remained secured even after the movement of a guard from a vertex to another. All components of the problem, including number of guards required, deployment of guards among various nodes, and movement strategies for guards to counter intruder attacks, were probed. For the sake of generality, guards were allowed to have different ranges from each other. Bounds on the number of guards required for the eternal security of various classes of graphs were also presented.

Robustness in multiagent systems, both from functional and structural perspectives, was studied in Chapter 6. Inter-relationship between structural and functional robustness was explored for networks implementing linear consensus dynamics, thereby proposing Kirchhoff index as a metric to quantify network robustness. Using Kirchhoff index, a systematic way to build robust network topologies was presented.

APPENDIX A

PROOF OF THEOREM 2.3.2

In order to prove Theorem 2.3.2, it is sufficient to prove that if G is a connected graph of minimum degree at least two with no induced subgraph isomorphic to $K_{1,6}$, and G is not isomorphic to C_4, C_7, $C_4 \bullet C_4$ or $K_{2,3}$, then G has a $(5,2)$-configuration.

This is true because of the following:

if s = even: let $s = 2s'$. Obtain a $(5,2)$-configuration of G. Repeat this process s' number of times while assigning distinct labels to vertices in each step. At the end of s' steps, at most $2s'$ labels are assigned to every v, i.e., $|f(v)| \leq 2s' = s$, and $5s' = \frac{5s}{2}$ distinct labels are available in the closed neighborhood of every v. Thus, a $(\frac{5s}{2}, s)$-configuration of G is obtained.

if s = odd: let $s - 1 = 2s'$, then $(\frac{5(s-1)}{2}, s-1)$-configuration can be obtained as above. Moreover, using the fact that every connected graph has a domination number of at least 2, it is possible to assign a single label to each vertex such that every vertex has at least two distinct labels in its closed neighborhood. Thus, for a given positive odd integer s, an (r,s)-configuration is possible with $r = \frac{5(s-1)}{2} + 2 = \frac{5s}{2} - \frac{1}{2} = \lfloor \frac{5s}{2} \rfloor$.

Thus, we conclude that proving Theorem 2.3.2 is equivalent of proving the following result.

Theorem A.0.1 *If G is a connected graph of minimum degree at least two with no induced subgraph isomorphic to $K_{1,6}$, and G is not isomorphic to C_4, C_7, $C_4 \bullet C_4$ or $K_{2,3}$, then G has a $(5,2)$-configuration.*

However, several preliminary results are required to prove Theorem A.0.1. We state and prove them in Section A.1. But, first we introduce some notations that will be used.

For the sake of brevity, let us define a *configuration* of a graph to mean a $(5,2)$-configuration. Thus, the conclusion of Theorem A.0.1 is equivalent of saying that G has a configuration. Moreover, we say that a graph is *configurable* if it admits a configuration. An (α,β)-*star* is a graph obtained by identifying one end of each of α number of paths of length one and β number of paths of length two as shown in Fig. A.1. Note that an $(\alpha,0)$-star is isomorphic to $K_{1,\alpha}$. We denote by $[5]^2$ the set of all two-element subsets of $\{1,2,3,4,5\}$. If G is a graph, $f : V(G) \to [5]^2$, and $v \in V(G)$, then we say that v *is satisfied* with respect to f if $\bigcup_{u\in\mathcal{N}[v]} f(u) = \{1,2,3,4,5\}$. When there is no danger of confusion, the reference to f will be omitted. The degree of a vertex v in a graph G is denoted by $\deg_G(v)$

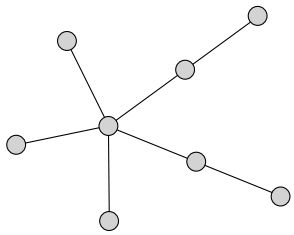


Figure A.1: The $(3,2)$-star graph.

A.1 Preliminary Lemmas

Lemma A.1.1 *Let $v_1v_2v_3v_4$ be a path of length three, and $f : \{v_1,v_4\} \to [5]^2$ with $f(v_1) \cap f(v_4)$ is nonempty. If $a,b \in \{1,2,3,4,5\} \setminus f(v_1)$, then f can be extended to $\{v_1,v_2,v_3,v_4\}$ in such a way that v_2 and v_3 are satisfied and $f(v_2) = \{a,b\}$.*

Proof. Without loss of generality, $f(v_1) = \{1,2\}$, $1 \in f(v_4)$, and $f(v_2) = \{a,b\} = \{3,4\}$. Then setting $f(v_3) = \{2,5\}$ completes the proof. ∎

Lemma A.1.2 *Let H and S be disjoint subgraphs of a graph G, and let $\alpha,\beta \geq 0$ be integers such that either $\alpha+3\beta \leq 9$ or $(\alpha,\beta) = (1,3)$. Let H be configurable and let S be either a path of length at least two or an (α,β)-star. If every vertex of S of degree*

one is adjacent to some vertex of H, then the subgraph of G induced by $V(H) \cup V(S)$ is configurable.

Proof. Let f be a configuration on H. First, suppose that $S = v_1v_2...v_k$ is a path of length at least two (so $k \geq 3$), and that the ends of S are adjacent to vertices x, y of H. Note that x and y may be the same vertex. There are three cases depending on the cardinality of $f(x) \cap f(y)$ and three cases depending on the residue of k modulo 3. Without loss of generality we may assume that $f(x) = f(y) = \{1, 2\}$, or $f(x) = \{1, 2\}$ and $f(y) = \{1, 3\}$, or $f(x) = \{1, 2\}$ and $f(y) = \{3, 4\}$. Then f can be extended to $V(H) \cup V(S)$ according to the following table, where t runs from 1 through $\lfloor k/3 \rfloor - 1$.

k (mod 3)	$f(x)$	$f(v_{3t+1})$	$f(v_{3t+2})$	$f(v_{3t+3})$	$f(v_{k-1})$	$f(v_k)$	$f(y)$
0	{1,2}	{1,3}	{4,5}	{2,3}	x	x	{1,2}
0	{1,2}	{3,4}	{1,5}	{2,4}	x	x	{1,3}
0	{1,2}	{3,4}	{1,5}	{1,2}	x	x	{3,4}
1	{1,2}	{3,4}	{1,5}	{2,5}	x	{3,4}	{1,2}
1	{1,2}	{3,4}	{1,5}	{2,5}	x	{3,4}	{1,3}
1	{1,2}	{3,5}	{1,4}	{1,2}	x	{3,5}	{3,4}
2	{1,2}	{3,4}	{1,5}	{1,2}	{3,4}	{1,5}	{1,2}
2	{1,2}	{3,4}	{2,5}	{1,2}	{3,4}	{2,5}	{1,3}
2	{1,2}	{3,4}	{1,5}	{2,4}	{1,3}	{2,5}	{3,4}

Now we assume that S is a (α, β)-star, where $\alpha + \beta \geq 3$, $\alpha + 3\beta \leq 9$, or $(\alpha, \beta) = (1, 3)$. Let $V(S) = \{w, x_i, y_j, z_j : 1 \leq i \leq \alpha, 1 \leq j \leq \beta\}$, $E(S) = \{wx_i, wy_j, y_jz_j : 1 \leq i \leq \alpha, 1 \leq j \leq \beta\}$, and x_i is adjacent to u_i, where u_i is in H, for all $1 \leq i \leq \alpha$, and z_j is adjacent to v_j, where v_j is in H, for all $1 \leq i \leq \beta$.

We say that u_i *forbids* the set $f(u_i)$ and that v_j *forbids* the three 2-element subsets of $[5] - f(v_j)$. We claim that there is an element of $[5]^2$ that is not forbidden by any

u_i or v_j. Indeed, this is clear if $\alpha + 3\beta \leq 9$. But if $\beta = 3$, then the vertices v_1, v_2, v_3 collectively forbid at most eight sets, and hence the claim holds even when $\alpha = 1$ and $\beta = 3$. We define $f(w)$ to be an element of $[5]^2$ that is not forbidden by any u_i or v_j. Furthermore, if $\beta = 0$ and $|\bigcup_{i=1}^{\alpha} f(u_i)| \leq 3$, then we choose $f(w)$ disjoint from every $f(u_i)$.

If $\beta \geq 1$, then we choose $f(x_i)$, $f(y_j)$ and $f(z_j)$ for $i = 1, 2, \ldots, \alpha$ and $j = 1, 2, \ldots, \beta - 1$ in such a way that the vertices x_i, y_j, z_j are satisfied. Then w sees at least three values under f since any neighbor of w already assigned a value does not have the exact same assignment as w. So by Lemma A.1.1 applied to the path $wy_\beta z_\beta v_\beta$ we can assign $f(y_\beta)$ and $f(z_\beta)$ in such a way that y_β, z_β and w are satisfied. This completes the case $\beta \geq 1$.

So we may assume $\beta = 0$. We assign $f(x_i)$ for $i = 1, 2, \ldots, \alpha$ such that x_i is satisfied, $f(x_i) \cap f(w) = \emptyset$, and, if possible, not all $f(x_i)$ are the same. Then w is satisfied, unless the sets $f(x_i)$ are all equal, and so from the symmetry we may assume that $f(w) = \{1, 2\}$ and $f(x_i) = \{3, 4\}$ for all $i = 1, 2, \ldots, \alpha$. But then the choice of $f(x_i)$ implies that $f(u_i) \subseteq \{1, 2, 5\}$, contrary to the choice of $f(w)$. ∎

Lemma A.1.3 *Let G be a graph, and let $P = xv_1v_2v_3y$ be a path in G.*

Proof. Let f be a configuration on H. We shall extend f to $V(G)$. If $f(x) = f(y)$, say $f(x) = \{1, 2\}$, then $H \setminus xy$ is also configurable, so we can extend f to $V(G)$ by Lemma A.1.2. So we may assume that $f(x) \neq f(y)$; that is, $|f(x) \cup f(y)| \geq 3$. Define $g : V(G) \to [5]^2$ by $g(v_1) = f(y)$, $g(v_3) = f(x)$, let $g(v_2)$ be a 2-element subset of $[5]^2$ containing $\{1, 2, 3, 4, 5\} \setminus (f(x) \cup f(y))$, and let $g(v) = f(v)$ for all $v \in V(G) \setminus \{v_1, v_2, v_3\}$. It is clear that g is a configuration on G. ∎

Lemma A.1.4 *Let H be C_4, C_7 or a configurable graph, and let u_0 be a vertex of H. Let G be a graph, where $V(G) = V(H) \cup \{u_i, w_j : 1 \leq i \leq k, 1 \leq j \leq m\}$ and*

$E(G) = E(H) \cup \{u_i u_{i+1}, u_k w_1, w_j w_{j+1}, w_m w_1 : 0 \leq i \leq k-1, 1 \leq j \leq m-1\}$ *for some nonnegative integer* k *and integer* m *with* $m \geq 3$. *Then* G *is configurable.*

Proof. By Lemma A.1.3 we may assume that $k = 0, 1$ or 2. Let C be the cycle $w_1 w_2 ... w_m w_1$. Since H is C_4, C_7 or a configurable graph, we may satisfy every vertex of H except possibly u_0 and u_0 is missing at most 2 colors. So we may assume $f(u_0) = \{1, 2\}$ and that u_0 is missing 3 and 4. Similarly we may choose f on C in such a way every vertex of C except possibly w_1 is satisfied, and that w_1 is missing at most 2 colors.

If $k = 0$ we choose f on C so that $f(w_1) = \{3, 4\}$ and the colors missing at w_1 are 1 and 2.

If $k = 1$, we choose f on C so that $f(w_1) = \{2, 5\}$ and the colors missing at w_1 are 3 and 4. We set $f(u_1) = \{3, 4\}$.

Finally, if $k = 2$, we choose f on C so that $f(w_1) = \{2, 3\}$ and the colors missing at w_1 are 1 and 5. We set $f(u_1) = \{3, 4\}$ and $f(u_2) = \{1, 5\}$. ∎

Lemma A.1.5 *Let* H *be a configurable graph, and let* f *be a configuration on* G. *If* G *is obtained from* H *by either*

- *adding a vertex* v *and two edges* vx *and* vy *to* H, *where* x, y *are vertices of* H *and* $f(x) \neq f(y)$, *or*
- *adding two vertices* u, v *and three edges* xu, uv, vy *to* H, *where* x, y *are vertices of* H *and* $f(x) \cap f(y) \neq \emptyset$,

then f *can be extended to* G.

Proof.

This is easy to verify. ∎

A graph G is said to be obtained from a graph H by *attaching* a path P if $V(G) = V(H) \cup V(P)$ and $E(G) = E(H) \cup E(P) \cup \{v_1 x, v_k y\}$, where v_1 and v_k are

the ends of P, and x, y are distinct vertices of H. A graph G is said to be obtained from a graph H by *adding* a path P if G is obtained from the disjoint union of H and P by identifying one end of P and x and identifying the other end of P and y, where x and y are distinct vertices of H.

Lemma A.1.6 *Let C be a cycle of length of five or six. If G is obtained from C by attaching a path of length two or three between two non-adjacent vertices in C, then G is configurable.*

Proof.

Let $C = v_1v_2...v_kv_1$, and P be the path in $G \setminus C$ where the end of P is adjacent to vertices u, v of C in G. If C is C_5, then we define a function $f : V(C) \to [5]^2$ by $f(v_i) = \{i, i+3\}$ for each $i = 1, 2, 3, 4, 5$, where the addition is modulo five. If C is C_6, then define $f(v_1) = \{1, 3\}, f(v_2) = \{2, 4\}, f(v_3) = \{1, 5\}, f(v_4) = \{2, 3\}, f(v_5) = \{1, 4\}, f(v_6) = \{2, 5\}$. So $f(x) \neq f(y)$ for all distinct vertices x, y in C, and $f(x) \cap f(y) \neq \emptyset$ for all non-adjacent two vertices x, y in C. Hence, f can be extended to G by Lemma A.1.5 since P is a path of length two or three. ∎

Lemma A.1.7 *Let x, y be vertices of a configurable graph H, let $C = v_1v_2...v_5v_1$ be a cycle of length five, and let $P = u_1u_2...u_p$ and $Q = w_1w_2...w_q$ be paths, where $p, q \in \{1, 2\}$. If G is the graph with $V(G) = V(H) \cup V(C) \cup V(P) \cup V(Q)$ and $E(G) = E(H) \cup E(C) \cup E(P) \cup E(Q) \cup \{xu_1, u_pv_1, yw_1, w_qv_3\}$, then G is configurable.*

Proof.

Let f be a configuration on H. We shall extend f to G. If $f(x) \cap f(y)$ is nonempty, say $1 \in f(x) \cap f(y)$, then let a, b are two distinct numbers in $\{1, 2, 3, 4, 5\} \setminus (f(x) \cup f(y))$, and define $f(v_1) = \{1, a\}$ and $f(v_3) = \{1, b\}$. If $f(x)$ is disjoint from $f(y)$, say $f(x) = \{1, 2\}$ and $f(y) = \{3, 4\}$, then define $f(v_1) = \{1, 3\}$ and $f(v_3) = \{1, 4\}$. By Lemma A.1.1 and Lemma A.1.5, there is a way to define f on $V(P) \cup V(Q) \cup \{v_2, v_4, v_5\}$ such that f is a configuration on G. ∎

Lemma A.1.8 *Let G be a graph obtained by attaching a path $P = v_1v_2...v_k$ to a cycle C with v_1 adjacent to x and v_k adjacent to y, for some vertices x, y in C, where $k \geq 3$. If G is not isomorphic to $C_4 \bullet C_4$, then G is configurable.*

Proof. If x is adjacent to y in C, then G is a cycle with a chord. So G is configurable when the cycle has length not four or seven. It is easy to check that G is configurable when the cycle has length four, and G is also configurable when the cycle has length seven by Lemma A.1.6. So we may assume that x is not adjacent to y in C. In other words, either x equals y, or x and y are non-adjacent.

If the length of C is not 4 or 7, then this lemma follows directly from Lemma A.1.2. So we may assume that the length of $C = u_1u_2...u_{|C|}u_1$ is four or seven. Also, we may assume that $3 \leq k \leq 5$ by Lemma A.1.3. Without loss of generality, we assume that $x = u_1$.

Case 1: $C = C_4$ and $x = y$. Then $k = 4$ or 5 since G is not isomorphic to $C_4 \bullet C_4$. So G is isomorphic to the graph obtained by attaching a path of order three to C_5 or C_6, and hence G is configurable by Lemma A.1.2.

Case 2: $C = C_4$ and $x \neq y$. We may assume that $y = u_3$. If $k = 3$ or 5, then $u_1v_1v_2...v_ku_3u_2u_1$ is a cycle of length six or eight, so it is configurable, and there is a configuration f on it. Then we can extend f to G by assigning that $f(u_3) = f(u_1)$, so G is configurable. If $k = 4$, then we define a configuration on G by $f(u_1) = \{1, 2\}, f(u_2) = \{3, 5\}, f(u_3) = \{3, 4\}, f(u_4) = \{2, 5\}, f(v_1) = \{1, 4\}, f(v_2) = \{3, 5\}, f(v_3) = \{2, 5\}, f(v_4) = \{1, 4\}$.

Case 3: $C = C_7$ and $x = y$. We may assume that $x = y = u_1$. If $k = 4$ or 5, then G is isomorphic to the graph obtained by attaching a path of order six to C_5 or C_6, so G is configurable by Lemma A.1.2. If $k = 3$, then we can define a configuration on G by $f(u_1) = \{1, 2\}, f(u_2) = \{3, 4\}, f(u_3) = \{1, 5\}, f(u_4) = \{2, 3\}, f(u_5) = \{1, 4\}, f(u_6) = \{2, 5\}, f(u_7) = \{3, 4\}, f(v_1) = \{1, 5\}, f(v_2) = \{3, 4\}, f(v_3) = \{2, 5\}$.

Case 4: $C = C_7$, $x = u_1$ and $y = u_6$. If $k = 3$ or 5, then G is isomorphic

to the graph obtained by attaching a path of order four to C_6 or C_8, so G is configurable by Lemma A.1.2. If $k = 4$, then we can define a configuration on G by $f(u_1) = \{1, 2\}, f(u_2) = \{3, 4\}, f(u_3) = \{3, 5\}, f(u_4) = \{1, 2\}, f(u_5) = \{4, 5\}, f(u_6) = \{3, 4\}, f(u_7) = \{3, 5\}, f(v_1) = \{1, 5\}, f(v_2) = \{3, 4\}, f(v_3) = \{2, 5\}, f(v_4) = \{1, 2\}$.

Case 5: $C = C_7$, $x = u_1$ and $y = u_5$. If $k = 4$ or 5, then G is isomorphic to the graph obtained by attaching a path of order three to C_8 or C_9, so G is configurable by Lemma A.1.2. If $k = 4$, then we can define a configuration on G by $f(u_1) = \{1, 2\}, f(u_2) = \{1, 3\}, f(u_3) = \{4, 5\}, f(u_4) = \{2, 3\}, f(u_5) = \{1, 2\}, f(u_6) = \{4, 5\}, f(u_7) = \{3, 4\}, f(v_1) = \{1, 5\}, f(v_2) = \{3, 4\}, f(v_3) = \{2, 5\}$. ∎

Lemma A.1.9 *The graph $K_{2,4}$ is configurable.*

Proof. Let $V(K_{2,4}) = \{x_1, x_2, y_1, y_2, y_3, y_4\}$, $E(K_{2,4}) = \{x_i y_j : 1 \leq i \leq 2, 1 \leq j \leq 4\}$. We define a configuration on $K_{2,4}$ by $f(x_1) = \{1, 2\}, f(x_2) = \{3, 4\}, f(y_1) = \{3, 5\}, f(y_2) = \{4, 5\}, f(y_3) = \{1, 5\}, f(y_4) = \{2, 5\}$. ∎

Lemma A.1.10 *If a graph G is obtained from $C_4 \bullet C_4$ or $K_{2,3}$ by attaching a path, then G is configurable.*

Proof. First, we assume that G obtained from $C_4 \bullet C_4$ by attaching a path $v_1 v_2 ... v_k$, where v_1 is adjacent to x, v_k is adjacent to y for some vertices x, y in $C_4 \bullet C_4$. We write the vertex set of $C_4 \bullet C_4$ as $\{u_1, u_2, u_3, v, w_1, w_2, w_3\}$, where $v u_1 u_2 u_3 v$ and $v w_1 w_2 w_3 v$ are the two cycles in $C_4 \bullet C_4$.

Case 1: $x = y$. By Lemma A.1.3, we may assume that $k = 2, 3$ or 4. If $x = y = u_1$, then G can be obtained from C_3 or C_5 by consecutively attaching a path of order three when $k = 2$ or 4, and G has a spanning subgraph which is obtained from two disjoint C_4's by attaching a path of order two when $k = 4$, so G is configurable by Lemma A.1.2 and Lemma A.1.4. Similarly, G is configurable if both x and y are u_3, w_1 or w_3. If $x = y = v_2$ and $k = 2$ or 4, then G can be obtained

from C_3 or C_5 by consecutively attaching a path of order three, so G is configurable by Lemma A.1.2. If $x = y = u_2$ and $k = 3$, then we define a configuration on G as $f(v) = \{3, 4\}, f(w_1) = \{1, 3\}, f(w_2) = \{2, 5\}, f(w_3) = \{1, 4\}, f(u_1) = \{4, 5\}, f(u_2) = \{1, 2\}, f(u_3) = \{2, 5\}, f(v_1) = \{1, 3\}, f(v_2) = \{4, 5\}, f(v_3) = \{2, 3\}$. Similarly, G is configurable if $x = y = w_2$. If $x = y = v$ and $k = 2$ or 4, then G can be obtained from C_3 or C_5 by consecutively attaching a path of order three. If $x = y = v$ and $k = 3$, then we define a configuration by $f(v) = \{1, 2\}, f(u_1) = \{1, 3\}, f(u_2) = \{4, 5\}, f(u_3) = \{2, 3\}, f(v_1) = \{1, 4\}, f(v_2) = \{3, 5\}, f(v_3) = \{2, 4\}, f(w_1) = \{1, 5\}, f(w_2) = \{3, 4\}, f(w_3) = \{2, 5\}$.

Case 2: $x \neq y$.By Lemma A.1.3, we may assume that $k = 0, 1, 2$. When $k = 0$, G is obtained by adding an edge xy to $C_4 \bullet C_4$, and it is easy to show that G is configurable. When $k = 1$, $x = v$, $y = u_2$, then define a configuration on G by $f(v) = \{1, 2\}, f(u_1) = \{4, 5\}, f(u_2) = \{3, 4\}, f(u_3) = \{1, 5\}, f(v_1) = \{2, 5\}, f(w_1) = \{1, 3\}, f(w_2) = \{4, 5\}, f(w_3) = \{2, 3\}$. Similarly, G is configurable if $k = 1$, $x = w_1$ and $y = w_3$. When $k = 1$ and x, y are not the case mentioned above, G has a spanning subgraph which is C_8, or it can be obtained from either C_5 by attaching a path, two disjoint C_4's by adding an edge, or C_5 by attaching paths of order one or two, so G is configurable by Lemma A.1.2, Lemma A.1.4, and Lemma A.1.5.

Now, we assume that G obtained from $K_{2,3}$ by attaching a path $v_1v_2...v_k$, where v_1 is adjacent to x, v_k is adjacent to y for some vertices x, y in $C_4 \bullet C_4$. We write $V(K_{2,3}) = \{u_1, u_2, w_1, w_2, w_3\}$ and $E(K_{2,3}) = \{u_iw_j : i = 1, 2, j = 1, 2, 3\}$.

Case 3: $x = y$. By Lemma A.1.3, we may assume that $k = 2, 3, 4$. Then G has a spanning subgraph which is obtained from either C_3 or C_5 by attaching a $(3, 0)$-star, or $C_4 \bullet C_4$ by attaching a path, or a cycle by attaching a C_4, so G is configurable by Lemma A.1.2, Lemma A.1.4, Case 1 and Case 2.

Case 4: $x \neq y$. By Lemma A.1.3, we may assume that $k = 0, 1, 2$. If $x = u_1$, $y = u_2$ and $k = 0$, then there is a configuration on G defined by $f(u_1) = \{1, 2\}, f(u_2) =$

$\{3,4\}, f(w_1) = f(w_2) = f(w_3) = \{1,5\}$. For other cases, G contains a subgraph which is isomorphic to $K_{2,4}$ or C_6, or it can be obtained from either C_3 by attaching a path of order three, $C_4 \bullet C_4$ by adding an edge, C_5 or C_6 by attaching paths of order one or two, so G is configurable by Lemma A.1.2, Lemma A.1.5, Lemma A.1.9, Case 1 and Case 2. ∎

Lemma A.1.11 *For every graph G, there is an orientation of $E(G)$ such that each vertex v has in-degree at least $\lfloor \deg_G(v)/2 \rfloor$.*

Proof. We proceed by induction on $|V(G)| + |E(G)|$. The lemma obviously holds for the null graph. e if $|E(G)| = 0$.

If v is an isolated vertex of G, then the lemma follows by induction applied to $G \backslash v$. If there is a vertex v in G of degree one, then, letting u be the unique neighbor of v, there is an orientation of $G \setminus uv$ such that the in-degree of each vertex x is at least $\lfloor \deg_{G\backslash\{uv\}}(x)/2 \rfloor$ by the induction hypothesis, and then we can obtain a desired orientation of G by orienting the edge uv from v to u.

So we may assume that G has minimum degree at least two, and hence G contains a cycle $C = v_1 v_2 ... v_k v_1$. By the induction hypothesis, there is an orientation of $G \backslash E(C)$ such that the in-degree of each vertex x is at least $\lfloor \deg_{G\backslash E(C)} /2 \rfloor$, and then we can obtain a desired orientation of G by orienting the edges of C to form a directed cycle. This completes the proof. ∎

It is to be noted that the proof in Lemma A.1.11 gives a linear-time algorithm to find such an orientation.

Lemma A.1.12 *Let H_1 and H_2 be graphs, let P be a path with at least one vertex, and let v_1 and v_2 be vertices of H_1 and H_2 respectively. Let G be the graph formed by taking the disjoint union of H_1, H_2, and P and identifying the first vertex of P with v_1 and the last vertex of P with v_2. Assume that H_1 and H_2 admit configurations f_1*

and f_2, respectively, and that for $i = 1, 2$ the configuration f_i satisfies every vertex of H_i except possibly v_i. If $|\bigcup_{u\in\mathcal{N}(v_1)} f_1(u)| \geq 4$ and $|\bigcup_{u\in\mathcal{N}(v_2)} f_2(u)| \geq 3$, then G is configurable.

Proof. Let f' be the function defined to be f_1 on H_1 and f_2 on H_2. Then f' is a configuration for G except on possibly v_1 and v_2 and P. Suppose $|V(P)| \leq 2$. Then we can permute the colors on f_2 so that v_1 and v_2 are satisfied, so we are done. If $|V(P)| = 3$, we may assume $f(v_1) = \{1, 2\}$ and v_1 is missing 3 and $f(v_2) = \{4, 5\}$ and v_2 is missing 3. Then we set $f(u) = \{1, 3\}$ where u is the middle vertex of P. If $|V(P)| = 4$, we apply Lemma A.1.1. If $|V(P)| \geq 5$, we can reduce to one of the previous cases by applying Lemma A.1.3. ∎

We are now ready to prove an important special case of Theorem A.0.1.

Lemma A.1.13 *Let G be a connected graph of maximum degree at most five and of minimum degree at least two with no two vertices of degree at least three adjacent. If G is not C_4, C_7, $C_4 \bullet C_4$ or $K_{2,3}$, then G is configurable.*

Proof. Let there be n vertices in G. Suppose that G is a minimum counterexample; that is, G is not configurable, but H is configurable for every graph H with $|V(H)| + |E(H)| < |V(G)| + |E(G)|$ that satisfies the conditions of the lemma.

We note first that we may assume G is 2-connected since otherwise we apply Lemma A.1.12, noting that each of the forbidden graphs except C_4 has the property that it admits a configuration except at possibly one vertex, v, with $|\bigcup_{u\in\mathcal{N}(v)} f(u)| = 4$. Since both graphs can't be C_4 (since $C_4 \bullet C_4$ is forbidden and two C_4's joined by a path are prevented by Lemma A.1.4), we are done.

Claim 1: G contains no C_4's.

Proof of Claim 1: Suppose there is a cycle $C = v_1v_2v_3v_4v_1$ of four vertices in G. If there is only one vertex, say v_1, in C of degree at least three in G, then it is a cut-vertex which is impossible.

Hence there are two vertices in C of degree at least three. We may assume that the two vertices are v_1 and v_3. Let $G' = G \setminus \{v_2\}$. If G' is configurable, then there is a configuration f on G', and we can extend f to G by assigning $f(v_2) = f(v_4)$, contradicting the assumption that G is not configurable. So G' is isomorphic to C_4, C_7, $C_4 \bullet C_4$ or $K_{2,3}$.

If G' is isomorphic to C_4, then G is isomorphic to $K_{2,3}$. If G' is isomorphic to C_7, then G is isomorphic to a graph obtained from C_4 by adding a path of length five, so G is configurable by Lemma A.1.8. If G' is isomorphic to $K_{2,3}$, then G is $K_{2,4}$, and it is configurable by Lemma A.1.9. So G' is isomorphic to $C_4 \bullet C_4$. Since v_4 is a vertex of degree two and it is a common neighbor of v_1 and v_3, we have that either v_1 or v_3 is the vertex of degree four in $C_4 \bullet C_4$. So G can be obtained from adding a path of length four to $K_{2,3}$, so G is configurable by Lemma A.1.10. □

Claim 2: If P is a path whose ends are of degree at least three in G and whose internal vertices are of degree two in G, then the number of internal vertices is at most two.

Proof of Claim 2: If the number of internal vertices of P is at least four, then consider the graph H which is obtained from G by replacing three consecutive degree two vertices in P by an edge. If H is configurable, G is also configurable by Lemma A.1.3. So H is C_4, C_7, $C_4 \bullet C_4$ or $K_{2,3}$. But in this case, G can be obtained from C_4 by attaching a path of order at least three, so G is configurable by Lemma A.1.8. If the number of internal vertices of P is three, then let H' be the graph obtained from P by deleting all internal vertices of P. Again, G is configurable by Lemma A.1.2 if H' is configurable. So H' is C_4, C_7, $C_4 \bullet C_4$ or $K_{2,3}$. However, G is configurable by Lemma A.1.8 and Lemma A.1.10 in this case. □

Claim 3: There are no induced (α, β)-stars S in G, where $\alpha + \beta \geq 3$, and $\alpha + 3\beta \leq 9$ or $(\alpha, \beta) = (1, 3)$, such that $G \setminus S$ has minimum degree at least two.

Proof of Claim 3: Suppose there is an induced (α, β)-star S, where $\alpha + \beta \geq 3$,

and $\alpha + 3\beta \leq 9$ or $(\alpha, \beta) = (1, 3)$, such that $G \setminus S$ has minimum degree at least two. Subject to this constraint, assume that $\alpha + \beta$ is as small as possible. Let $G' = G \setminus S$, and $M_1, M_2, ..., M_k$ be components of G'. If every component of G' is configurable, then G is configurable by Lemma A.1.2. So there is a component of G' which is not configurable, and hence this component is isomorphic to C_4, C_7, $C_4 \bullet C_4$ or $K_{2,3}$ by the minimality of G. But G contains no C_4's by Claim 1, so the component is isomorphic to C_7. Without loss of generality, we may assume that M_1 is isomorphic to C_7 and write $M_1 = v_1v_2...v_7v_1$.

If M_1 contains exactly one vertex of degree at least three in G, then G is configurable by Lemma A.1.4, a contradiction. If M_1 contains exactly two vertices of degree at least three in G, then there is a path of length at least four whose ends are of degree at least three in G and whose internal vertices are of degree two in G, contradicting Claim 2. Hence there are three vertices in M_1 of degree at least three in G, and we may assume that they are v_1, v_3, v_5. Furthermore, if all v_1, v_3, v_5 have degree at least four in G, then $\alpha + \beta \geq 6$ and G contains a C_4, contradicting Claim 1. Therefore, at least one of v_1, v_3, v_5, say x, has degree three in G. Note that there is an (α, β)-star with center x and $\alpha + \beta = 3$ such that the graph obtained from G by deleting this (α, β)-star is still of minimum degree at least two, so S must also have that $\alpha + \beta = 3$ by the minimality of $\alpha + \beta$. So G' is C_7 as $\alpha + \beta = 3$. In other words, $(\alpha, \beta) = (0, 3), (1, 2), (2, 1)$ or $(3, 0)$.

If $(\alpha, \beta) = (3, 0)$, then G can be obtained from C_6 by attaching a $(2, 1)$-star, so G is configurable by Lemma A.1.2. So this is not a $(3, 0)$-star. If $(\alpha, \beta) = (0, 3)$, then G is configurable since it can be obtained from C_8 by attaching a $(1, 2)$-star. If $(\alpha, \beta) = (1, 2)$, then G is configurable since G can be obtained from C_8 by attaching either a $(2, 1)$-star or $(3, 0)$-star. So $(\alpha, \beta) = (2, 1)$. Let $V(S) = \{a, b, c, d_1, d_2\}$ and $E(S) = \{ab, ac, ad_1, d_1d_2\}$. If d_2 is adjacent to v_1 or v_5, then G is configurable since it can be obtained from C_6 by attaching a $(1, 2)$-star. Thus, d_2 is adjacent to v_3. Hence, there is

a configuration on G defined as $f(v_1) = \{1, 2\}, f(v_2) = \{4, 5\}, f(v_3) = \{1, 3\}, f(v_4) = \{4, 5\}, f(v_5) = \{1, 2\}, f(v_6) = \{3, 4\}, f(v_7) = \{3, 5\}, f(a) = \{1, 3\}, f(b) = f(c) = \{4, 5\}, f(d_1) = \{2, 5\}, f(d_2) = \{2, 4\}$. This proves Claim 3. □

Claim 4: G contains no C_6's with exactly two diagonal vertices of degree at least three.

Proof of Claim 4: Let $C = v_1v_2...v_6v_1$ be a cycle of order six with v_1 and v_4 the two vertices of degree at least three in G. Since G has no adjacent vertices whose degrees are at least three, v_5 and v_6 have degree two in G. Let G' be the graph obtained by deleting v_5, v_6 from G, so G' is a graph of minimum degree at least two, maximum degree at most five, and there are no adjacent vertices whose degrees are at least three. If G' is not configurable, then G' is C_4, C_7, $C_4 \bullet C_4$ or $K_{2,3}$ by the minimality of G. However, G contains no C_4's, so G' is C_7 and it contains at most two vertices whose degrees in G are at least three. Hence, there is a path of order at least five whose internal vertices are of degree two, which contradicts to Claim 2. Consequently, G' is configurable and there is a configuration f on G', and we can extend f to G by defining $f(v_5) = f(v_3)$ and $f(v_6) = f(v_2)$. □

We now construct a configuration on G. Contruct a graph H as follows: the vertices of H are the vertices of degree at least three in G, and xy is an edge in H if x and y have a common neighbor in G.

Claim 5: The maximum degree of H is at most two.

Proof of Claim 5: Suppose there is a vertex x of degree at least three in H. Let $x_1, x_2, ..., x_k$ be the vertices of degree at least three such that there exist x-x_i paths of length two or three. Then the internal vertices of those x-x_i paths together with x form an (α, β)-star S with $\alpha \geq 3$. On the other hand, β is at most two since G is of maximum degree at most five. So S is an (α, β)-star with $\alpha + 3\beta \leq 9$. By Claim 3, $G \setminus S$ is not of minimum degree at least two. Therefore, the degree of x_i in $G \setminus S$ is at most one, for some $i = 1, 2, ..., k$. Since G contains no C_4's and C_6 with exactly

two diagonal vertices of degree at least three in G, the degree of x_i is exactly three. Thus, there is an (α', β')-star S' centered at x_i with $\alpha' + 3\beta' \leq 9$ such that $G \setminus S'$ is of minimum degree two since $\alpha \geq 3$, which contradicts Claim 3. Hence, the maximum degree of H is at most two. $\square$

By Claim 5, H is a disjoint union of isolated vertices, paths and cycles. Let H^2 be the graph obtained by adding edges xy to H for each pair of two vertices x, y which have distance exactly two between them in H, and then deleting multiple edges and loops. Therefore, H^2 has maximum degree at most four. Let H' be the graph that is obtained by deleting an edge which is in H^2 but not in H from each component of H^2 isomorphic to K_5. Hence, H' is 4-colorable by Brooks' Theorem. Let $c : V(H') \to \{1, 2, 3, 4\}$ be a proper 4-coloring of H' such that $c(v) = 1$ for each isolated vertex v in H. Note that H^2 contains a component which is isomorphic to K_5 if and only if the component in H is isomorphic to C_5.

Define a function $f : V(H) \to [5]^2$ as $f(v) = \{c(v), 5\}$ for every vertex v in H. Let U be the set of vertices u such that u is a common neighbor of two vertices of degree at least three in G. Since no two vertices of degree at least three are adjacent, every vertex in U is of degree two in G. Now, we shall extend f to $V(H) \cup U$ by defining $f(u) = \{1, 2, 3, 4, 5\} \setminus (f(x) \cup f(y))$ for each vertex u in U, where x, y are the two neighbors of u in G. Note that if x, y are the two neighbors of a vertex u in U, then $c(x) \neq c(y)$ since H' contains all edges in H, so $|f(x) \cup f(y)| = 3$, and f is well-defined on $V(H) \cup U$. It is clear that $\bigcup_{w \in \mathcal{N}[u]} f(u) = \{1, 2, 3, 4, 5\}$ for each $u \in U$. Furthermore, if v is a vertex with degree at least 2 in H, and v is not in a component of H isomorphic to C_5, then neighbors of v in H receive different colors under c, so u is satisfied. Similarly, for each component of H which is isomorphic to C_5, there is a vertex w such that $|\bigcup_{u \in \mathcal{N}[w] \cap (V(H) \cup U)} f(u)| = 4$ and each other vertex is satisfied.

Let W be the set of vertices w that are not satisfied. So each vertex in W is either

an isolated vertex in H, an end of a maximal path in H, or a vertex in a component of H which is isomorphic to C_5. Let $X = \{w \in W : w \text{ is an isolated vertex in } H\}$, and let Y be the set $W \setminus X$. Notice that $|\bigcup_{u \in \mathcal{N}[w] \cap (V(H) \cup U)} f(u)| = 4$ when w is in Y. Now construct a graph L, where $V(L)$ is equal to $V(H)$, and two vertices x, y in L are adjacent if there is a x-y path of length three in G. Note that since no vertices of degree at least three are adjacent, the internal vertices of every x-y path of length three in G are of degree two for each $xy \in E(L)$.

Claim 6: If w is in X, then the degree of w in L is at least four. If w is in Y, then the degree of w in L is at least two.

Proof of Claim 6: Let w be a vertex in $X \cup Y$. Let $x_1, x_2, ..., x_k$ be vertices of degree at least three in G such that there are w-x_i paths in G of length two or three for each $i = 1, 2, ..., k$. Then the internal vertices of those w-x_i paths together with w form an (α, β)-star S.

Suppose $w \in X$. Then $\alpha = 0$ and there is at most one path between w and each x_i since otherwise we violate Claim 4. But then $G \backslash S$ has minimum degree 2, so by Claim 3, $\beta \geq 4$, so the degree of w in L is at least four.

Suppose $w \in Y$ and that $\beta \leq 1$. If w was not in a C_5 in H, then $\alpha = 1$, so the degree of w is only 2. So we must have that w was in a C_5 in H, so $\alpha = 2$. Removing S must create a vertex of degree 1 by Claim 3, say x_1. So x_1 must have degree 3 and be part of a 5-cycle with w, C. Since w is in a C_5 in H, we must have that x_1 has a path of length two to another vertex of degree at least three and that the graph H obtained by removing C and the two adjacent degree 2 vertices is connected and of minimum degree 2. If it were configurable, then by Lemma A.1.7, G would be as well, so H must be C_7 which is impossible since it has at least one degree 3 vertex since G has at least 5 degree 3 vertices since w is in a C_5 in H. $\square$

By Lemma A.1.11, L then has an orientation in which each vertex of X has in-degree at least 2 and every vertex in Y has in-degree at least 1. We use this to extend

f to satisfy every vertex in G. Each edge in L corresponds to a path of length 3, x, v_1, v_2, y in G (where x is the tail of the edge in L). For each of these paths, let a, b be two colors not in $\bigcup_{u \in \mathcal{N}(x)} f(u)$ (if that many colors exist, otherwise arbitrarily add colors not in $f(x)$). Then assign $f(v_1) = (a, b)$ and $f(v_2)$ as given by Lemma A.1.1.

At the end of this process each vertex of degree 2 is satisfied. Each vertex not in X or Y was already satisfied. Each vertex in X was the tail of two edges in L, so sees up to 4 new colors, so is certainly satisfied. Each vertex in Y was only missing at most 2 colors, but was the tail of at least one edge in L, so is now satisfied. ∎

A.2 Main Theorems

Now, we prove Theorem A.0.1 (thus, proving Theorem 2.3.2).

A.2.1 Proof of Theorem A.0.1.

We proceed by induction on $|V(G)| + |E(G)|$. If the order of G is 3, then G is C_3 and G is configurable. If the order of G is 4, 5, 6 or 7, but G is not C_4, C_7, $C_4 \bullet C_4$ or $K_{2,3}$, then G is configurable by Lemmas A.1.6, A.1.8, and A.1.10. So we may assume that G at least 8 vertices.

Suppose there is a vertex v of degree two in G such that v is in a $C_4 = vabcv$ with degree of b also two. If the degree of a is also 2, then G is obtained by adding a four-path to a forbidden graph which is impossible by Lemmas A.1.8 and A.1.10. So we may assume that a and c have degree at least 3.

So we consider $G \setminus v$. If it has a configuration f, then G is configurable since we may extend f to $V(G)$ by assigning $f(v) = f(b)$. As the order of G is at least 8, $G \setminus v$ is not configurable only if $G \setminus v$ is C_7 or $C_4 \bullet C_4$. However, it is not hard to see that if $G \setminus v$ is C_7 or $C_4 \bullet C_4$, then G can be obtained either from C_5 or C_6 by attaching paths of order one or two, or from C_3 by consecutively attaching a path of order at least three, so G is configurable by Lemmas A.1.5 and A.1.8. Hence, we may assume that no four cycle has two vertices of degree two opposite one another.

Suppose there were three vertices x, y, z in G such that x, y, z forms a triangle in G and the degree of y and z were exactly two. By the induction hypothesis, Lemma A.1.5 and Lemma A.1.10, G is configurable if $G \setminus \{y, z\}$ is not C_4 or C_7. But if $G \setminus \{y, z\}$ is C_4 or C_7, then G can be obtained from C_3 by attaching a path with order at least three, so G is still configurable by Lemma A.1.8. Hence, we may assume that G has no triangles with two vertices of degree three.

Let G' be a spanning subgraph of G such that the minimum degree of G' is at least two and satisfies the following:

1. $|E(G')|$ is as small as possible,
2. subject to that, the number of triangles in G' is as small as possible, and
3. subject to that, the number of components in G' which are isomorphic to $C_4 \bullet C_4$ or $K_{2,3}$ is as small as possible.

We shall prove the following claims.

Note that by the minimality of $E(G')$, there are no two vertices of degree at least three adjacent to one another.

Claim 1: The maximum degree of G' is at most five.

Proof of Claim 1: Suppose that there is a vertex v of degree at least six in G'. As G is $K_{1,6}$-free, there are two vertices x, y adjacent to v in G' with x adjacent to y in G. Since the degree of v is at least three, x and y must have degree two in G'. If $xy \notin E(G')$, then the graph obtained by deleting xv, yv from G' and then adding xy into G' is still a spanning subgraph of G with minimum degree at least two, but it has fewer edges. So $xy \in E(G')$, in other words, v, x, y form a triangle in G'. Since x, y, v form a triangle in G and the degree of v is at least three, at least one of x and y has degree at least three in G. We may assume that the degree of x in G is at least three, and u is a neighbor of x in G other than y and v. As $xy, vx \in E(G')$ and the degree

of x is two in G', $xu \notin E(G')$. So the graph obtained by deleting xv and adding xu has the same number of edges but it has fewer triangles than G', a contradiction. □

Claim 2: Let Q be a component of G' isomorphic to $C_4 \cdot C_4$ or $K_{2,3}$. Let u be a vertex of degree two adjacent to a vertex of degree at least three in Q, v. Then there are no edges between u and $V(G) \setminus V(Q)$.

Proof of Claim 2: Let $e = uv$. Then the graph obtained by deleting e and adding uv to G' is still of minimum degree two, and it has the same number of edges and the same number of triangles as G', but it has fewer components isomorphic to $C_4 \cdot C_4$ or $K_{2,3}$, contradicting the minimality of G'. □

Since every component of G' is a connected graph of minimum degree at least two and of maximum degree at most five, and no vertices of degree at least three in G' are adjacent to one another, every component of G' is configurable except those that are isomorphic to C_4, C_7, $C_4 \cdot C_4$, or $K_{2,3}$ by Lemma A.1.13. Also, it is clear that if a graph contains C_4, C_7, $C_4 \cdot C_4$ or $K_{2,3}$ as a spanning subgraph but not as an induced subgraph, then it is also configurable.

Let $R_1, R_2, ...R_k$ be components which are not configurable. Then the subgraph of G induced by each $V(R_i)$ is C_4, C_7, $C_4 \bullet C_4$ or $K_{2,3}$. By Claim 2, edges incident to a degree two vertex of R_i must be incident to another vertex in the same R_i, so every vertex of degree two in R_i is also of degree two in G. But we already assume that no C_4 contains two opposite vertices of degree two, so R_i must be isomorphic to C_4 or C_7.

Now, we show that G is configurable. Let H be the maximum configurable subgraph of G induced by the union of components of G'. If $H \neq G$, then let C be a component of G' not in H. Then C is either C_4 or C_7 and has an edge either to H or to another C_4 or C_7. In either case it is easy to extend the configuration to include C (and possibly an additional cycle) which contradicts the maximality of H. So G is configurable. ■

Note that our proof gives a polynomial time algorithm to find a configuration of an n-vertex graph G if G is a $K_{1,6}$-free graph of minimum degree at least two, and no component of G is isomorphic to C_4, C_7, $C_4 \bullet C_4$, or $K_{2,3}$.

APPENDIX B

PROOF OF THEOREM 6.2.2

In order to prove Theorem 6.2.2, we need to prove the following preliminary Lemmas B.1.1 – B.1.9.

B.1 Preliminary Lemmas

Lemma B.1.1 *Let $G = (\mathcal{F}_3)^k$, then,*

$$K_f(G) \;=\; \frac{2}{21}k(92k-19) \tag{84}$$

Proof. Let us label the vertices in $(\mathcal{F}_3)^k$ as shown in the Fig. B.1(a). Here, $r_{a,\alpha} = r_{d,\alpha} = 13/21$ and $r_{b,\alpha} = r_{c,\alpha} = 10/21$. Also, we can write $K_f(G)$ as

$$\begin{aligned}
K_f(G) =&(k)K_f(\mathcal{F}_3) \;+\; \left(\sum_{i\in\mathcal{F}_3, j\in\mathcal{F}_3} r_{i,j}\right)\;(1+2+\cdots+(k-1))\\
=&(k)K_f(\mathcal{F}_3) \;+\; \left(\sum_{j\in\mathcal{F}_3} r_{a,j} + \sum_{j\in\mathcal{F}_3} r_{b,j} + \sum_{j\in\mathcal{F}_3} r_{c,j} + \sum_{j\in\mathcal{F}_3} r_{d,j}\right)\;(1+2+\cdots+(k-1))\\
=&(k)K_f(\mathcal{F}_3) \;+\; 2\left(\sum_{j\in\mathcal{F}_3} r_{a,j} + \sum_{j\in\mathcal{F}_3} r_{b,j}\right)\;(1+2+\cdots+(k-1))\\
=&(k)K_f(\mathcal{F}_3) \;+\; k(k-1)\left(\frac{98}{21} \;+\; \frac{86}{21}\right)
\end{aligned}$$

Since, $K_f(\mathcal{F}_3) = 146/21$, so, we get,

$$\begin{aligned}
K_f(G) =&\frac{146k}{21} \;+\; \frac{184}{21}k(k-1)\\
=&\frac{2}{21}k(92k-19)
\end{aligned}$$

■

Lemma B.1.2 *Let $H = (\mathcal{F}_1)^k \bullet (\mathcal{F}_3)^\ell$, then,*

$$K_f(H) \;=\; \frac{2}{21}\ell(92\ell - 19) \;+\; \frac{2}{3}k(4k-1) \;+\; \frac{204}{21}k\ell \tag{85}$$

Proof. Let us label the vertices in $(\mathcal{F}_1)^k$ as shown in the Fig. B.1(b). Here, $r_{a,\alpha} = r_{b,\alpha} = 2/3$. Also,

$$\begin{aligned} K_f(H) \;&= K_f((\mathcal{F}_1)^k) \;+\; K_f((\mathcal{F}_3)^\ell) \;+ \sum_{i\in(\mathcal{F}_1)^k, j\in(\mathcal{F}_3)^\ell} r_{i,j} \\ &= K_f((\mathcal{F}_1)^k) \;+\; K_f((\mathcal{F}_3)^\ell) \;+\; k\ell\left(\sum_{j\in(\mathcal{F}_3)^\ell} r_{a,j} \;+ \sum_{j\in(\mathcal{F}_3)^\ell} r_{b,j}\right) \\ &= K_f((\mathcal{F}_1)^k) \;+\; K_f((\mathcal{F}_3)^\ell) \;+\; 2k\ell\left[4(\frac{2}{3}) \;+\; 2(\frac{13}{21}) \;+\; 2(\frac{10}{21})\right] \end{aligned}$$

Now, using (46) and (84), we get,

$$K_f(H) \;=\; \frac{2}{21}\ell(92\ell - 19) \;+\; \frac{2}{3}k(4k-1) \;+\; \frac{204}{21}k\ell$$

■

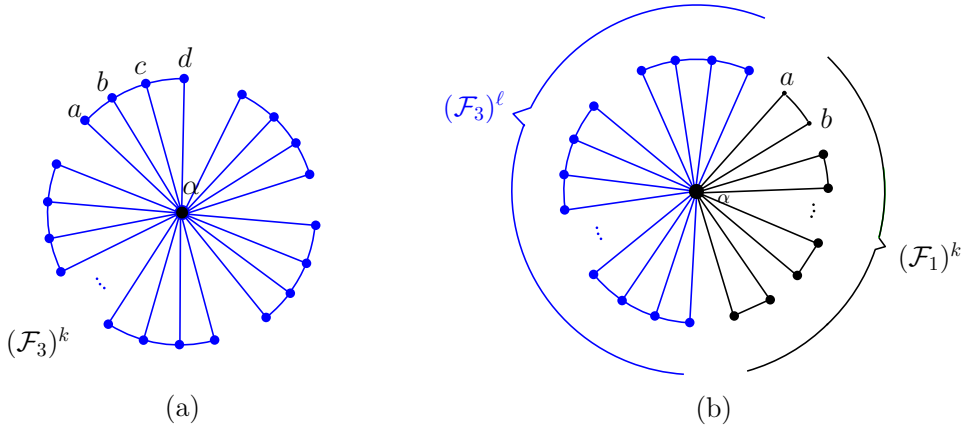


Figure B.1: (a) Labeling of $(\mathcal{F}_3)^k$ for the proof of Lemma B.1.1. (b) Labeling of $H = (\mathcal{F}_1^k) \bullet (\mathcal{F}_3)^\ell$, for the proof of Lemma B.1.2.

Lemma B.1.3 *Let $H = ((\mathcal{F}_1)^k \bullet (\mathcal{F}_3)^\ell) \bullet \mathcal{F}_7$, then,*

$$K_f(H) \;=\; \frac{2}{21}\ell(92\ell + 332.0612) \;+\; \frac{2}{3}k(4k + 26.9336) \;+\; \frac{204}{21}k\ell \;+\; 27.7994 \tag{86}$$

Proof. Let us label the vertices in H as shown in the Fig. B.2. Also, we can write the $K_f(H)$ as below,

$$K_f(H) = K_f((\mathcal{F}_1)^k \bullet (\mathcal{F}_3)^\ell)) + K_f(\mathcal{F}_7) + + \sum_{i\in\mathcal{F}_7, j\in(\mathcal{F}_1)^k} r_{i,j} + \sum_{i\in\mathcal{F}_7, j\in(\mathcal{F}_3)^\ell} r_{i,j} \quad (87)$$

Firstly, we will compute $\sum_{i\in\mathcal{F}_7, j\in(\mathcal{F}_1)^k} r_{i,j}$. Note that for $\mathcal{F}_7$, we have, $r_{a,\alpha} = r_{h,\alpha}$, $r_{b,\alpha} = r_{g,\alpha}$, $r_{c,\alpha} = r_{f,\alpha}$, $r_{d,\alpha} = r_{e,\alpha}$. Also, $r_{x,j} = r_{x,\alpha} + r_{\alpha,j}$, where $x \in \{a, b, c, d, e, f, g, h\}$ and j is a vertex from $(\mathcal{F}_1)^k$. So,

$$\sum_{i\in\mathcal{F}_7, j\in(\mathcal{F}_1)^k} r_{i,j} = 2\left[\sum_{j\in(\mathcal{F}_1)^k} r_{a,j} + \sum_{j\in(\mathcal{F}_1)^k} r_{b,j} + \sum_{j\in(\mathcal{F}_1)^k} r_{c,j} + \sum_{j\in(\mathcal{F}_1)^k} r_{d,j}\right]$$

Noting that $r_{\alpha,j} = 2/3$, where j is in $(\mathcal{F}_1)^k$. Also using $r_{x,\alpha}$ values, where $x \in \{a, b, c, d\}$, we get,

$$\sum_{i\in\mathcal{F}_7, j\in(\mathcal{F}_1)^k} r_{i,j} = 2\left[\sum_{j\in(\mathcal{F}_1)^k} 1.2847 + \sum_{j\in(\mathcal{F}_1)^k} 1.1388 + \sum_{j\in(\mathcal{F}_1)^k} 1.1176 + \sum_{j\in(\mathcal{F}_1)^k} 1.1145\right]$$

Since, there are $2k$ vertices in $(\mathcal{F}_1)^k$, thus, we get,

$$\sum_{i\in\mathcal{F}_7, j\in(\mathcal{F}_1)^k} r_{i,j} = 2(2k)\,[1.2847 + 1.1388 + 1.1176 + 1.1145] = (18.6224)k \quad (88)$$

Now, we compute $\sum_{i\in\mathcal{F}_7, j\in(\mathcal{F}_3)^\ell} r_{i,j}$

$$\begin{aligned}\sum_{i\in\mathcal{F}_7, j\in(\mathcal{F}_3)^\ell} r_{i,j} &= \ell\left[2\sum_{j\in\mathcal{F}_3} r_{a,j} + 2\sum_{j\in\mathcal{F}_3} r_{b,j} + 2\sum_{j\in\mathcal{F}_3} r_{c,j} + 2\sum_{j\in\mathcal{F}_3} r_{d,j}\right] \\ &= 2\ell\ [4.6625 + 4.0789 + 3.9941 + 3.9817] \\ &= (33.4344)\ell\end{aligned} \quad (89)$$

Now, using $K_f(\mathcal{F}_7) = 27.7994$, and inserting (85), (88) and (89) in (87), we get,

$$K_f(H) \;=\; \frac{2}{21}\ell(92\ell + 332.0612) \;+\; \frac{2}{3}k(4k + 26.9336) \;+\; \frac{204}{21}k\ell \;+\; 27.7994$$

■

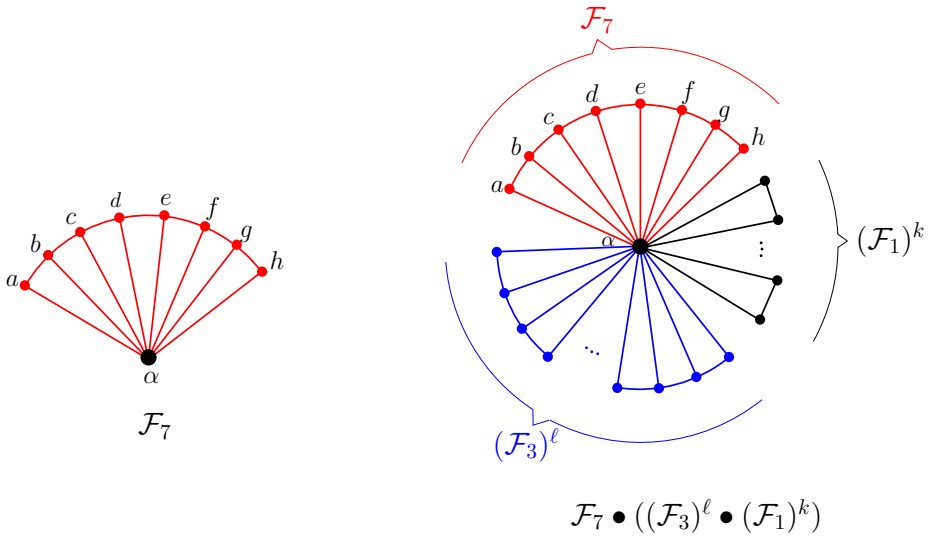


Figure B.2: (a) Labeling of $((\mathcal{F}_1)^k \bullet (\mathcal{F}_3)^\ell) \bullet \mathcal{F}_7$ for the proof of Lemma B.1.3.

Lemma B.1.4 *Let* $H = ((\mathcal{F}_1)^k \bullet (\mathcal{F}_3)^\ell) \bullet \mathcal{F}_5$, *then,*

$$K_f(H) \;=\; \frac{2}{21}\ell(92\ell + 248.5033) \;+\; \frac{2}{3}k(4k + 20.2502) \;+\; \frac{204}{21}k\ell \;+\; 15.5833 \quad (90)$$

Proof. Kirchhoff index of given H can be written as,

$$K_f(H) \;=\; K_f((\mathcal{F}_1)^k \bullet (\mathcal{F}_3)^\ell)) \;+\; K_f(\mathcal{F}_5) \;+\; \sum_{i\in\mathcal{F}_5, j\in(\mathcal{F}_1)^k} r_{i,j} \;+\; \sum_{i\in\mathcal{F}_5, j\in(\mathcal{F}_3)^\ell} r_{i,j} \quad (91)$$

Using exactly the same procedure as in Lemma B.1.3, we compute,

$$\sum_{i\in\mathcal{F}_5, j\in(\mathcal{F}_1)^k} r_{i,j} \;=\; (14.1668)k \quad (92)$$

and,

$$\sum_{i\in\mathcal{F}_5, j\in(\mathcal{F}_3)^\ell} r_{i,j} \;=\; (25.4765)\ell \quad (93)$$

Now using $K_f(\mathcal{F}_5) = 15.5833$, and inserting (85), (92), and (93) in (91), we get

$$K_f(H) \;=\; \frac{2}{21}\ell(92\ell + 248.5033) \;+\; \frac{2}{3}k(4k + 20.2502) \;+\; \frac{204}{21}k\ell \;+\; 15.5833$$

Thus, we get the desired result. $H = ((\mathcal{F}_1)^k \bullet (\mathcal{F}_3)^\ell) \bullet \mathcal{F}_5$ is shown in the Fig. B.3.

■

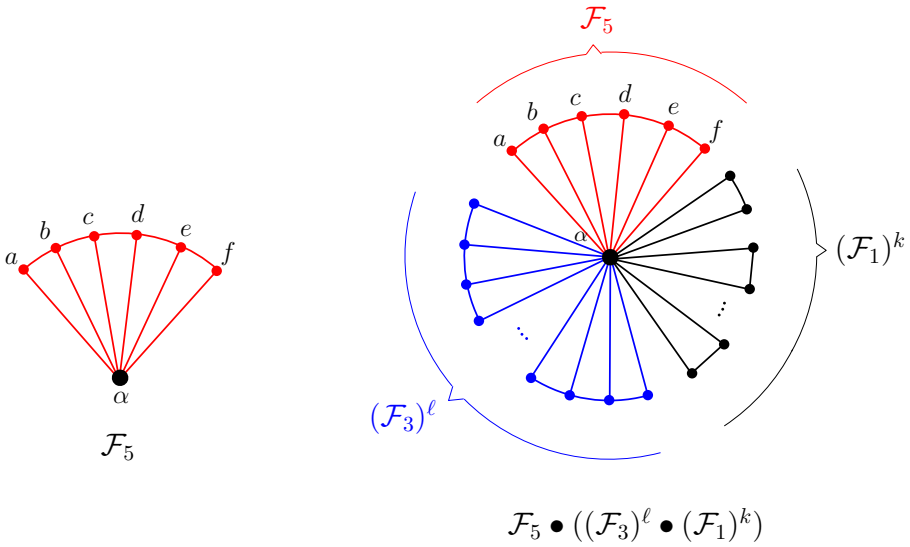


Figure B.3: $H = ((\mathcal{F}_1)^k \bullet (\mathcal{F}_3)^\ell) \bullet \mathcal{F}_5$ for the Lemma B.1.4.

Lemma B.1.5 *Let $\mathcal{M}_1$ be a graph as shown in the Fig. B.4, and $H = ((\mathcal{F}_1)^k \bullet (\mathcal{F}_3)^\ell) \bullet \mathcal{M}_1$, then,*

$$K_f(H) \;=\; \frac{2}{21}\ell(92\ell + 333.3653) \;+\; \frac{2}{3}k(4k + 27.0262) \;+\; \frac{204}{21}k\ell \;+\; 28.0779 \quad (94)$$

Proof. We can write $K_f(H)$ as,

$$K_f(H) \;=\; K_f((\mathcal{F}_1)^k \bullet (\mathcal{F}_3)^\ell) \;+\; K_f(\mathcal{M}_1) \;+\; \sum_{i \in (\mathcal{F}_1)^k, j \in \mathcal{M}_1} r_{i,j} \;+\; \sum_{i \in (\mathcal{F}_3)^\ell, j \in \mathcal{M}_1} r_{i,j} \quad (95)$$

Let us label the vertices in $\mathcal{M}_1$ as shown in the Fig. B.4. Then, we compute the following,

$$r_{a,\alpha} = 0.6147;\ \ r_{b,\alpha} = 0.4589;\ \ r_{c,\alpha} = 0.3680;\ \ r_{d,\alpha} = 0.5920;$$

$$r_{e,\alpha} = 0.4361;\ \ r_{f,\alpha} = 0.4491;\ \ r_{g,\alpha} = 0.4719;\ \ r_{h,\alpha} = 0.6180;$$

This gives us $\sum_{j\in\mathcal{M}_1} r_{j,\alpha} = 4.0087$. Now, we will calculate $\sum_{i\in(\mathcal{F}_1)^k, j\in\mathcal{M}_1} r_{i,j}$,

$$\begin{aligned}\sum_{i\in(\mathcal{F}_1)^k, j\in\mathcal{M}_1} r_{i,j} &= k\left(\sum_{i\in\mathcal{F}_1, j\in\mathcal{M}_1} r_{i,j}\right)\\ &= 2k\left[8\left(\frac{2}{3}\right) + \sum_{j\in\mathcal{M}_1} r_{j,\alpha}\right] = (18.684)k\end{aligned} \tag{96}$$

Similarly, we calculate $\sum_{i\in(\mathcal{F}_3)^\ell, j\in\mathcal{M}_1} r_{i,j}$ as following,

$$\begin{aligned}\sum_{i\in(\mathcal{F}_3)^\ell, j\in\mathcal{M}_1} r_{i,j} &= \ell\left(\sum_{i\in\mathcal{F}_3, j\in\mathcal{M}_1} r_{i,j}\right)\\ &= 2\ell\left[(8)\left(\frac{13}{21}\right) + \sum_{j\in\mathcal{M}_1} r_{j,\alpha} + (8)\left(\frac{10}{21}\right) + \sum_{j\in\mathcal{M}_1} r_{j,\alpha}\right] = (33.5586)\ell\end{aligned} \tag{97}$$

After calculating $K_f(\mathcal{M}_1) = 28.0779$, and inserting (85), (96), and (97) in (95), we get

$$K_f(H) = \frac{2}{21}\ell(92\ell + 333.3653) + \frac{2}{3}k(4k + 27.026) + \frac{204}{21}k\ell + 28.0779$$

■

Let us define graphs, $\mathcal{M}_2,\ \mathcal{M}_3,\ \mathcal{M}_4,\ \mathcal{M}_5$ as shown in the Fig. B.5.

Now, using exactly the same procedures as in Lemma B.1.3, Lemma B.1.5, following Lemmas can be proved.

Lemma B.1.6 *Let $H = \mathcal{M}_2 \bullet ((\mathcal{F}_1)^k \bullet (\mathcal{F}_3)^\ell)$, as shown in the Fig. B.6. Then,*

$$K_f(H) = \frac{2}{21}\ell(92\ell + 334.8522) + \frac{2}{3}k(4k + 27.1323) + \frac{204}{21}k\ell + 28.3972 \tag{98}$$

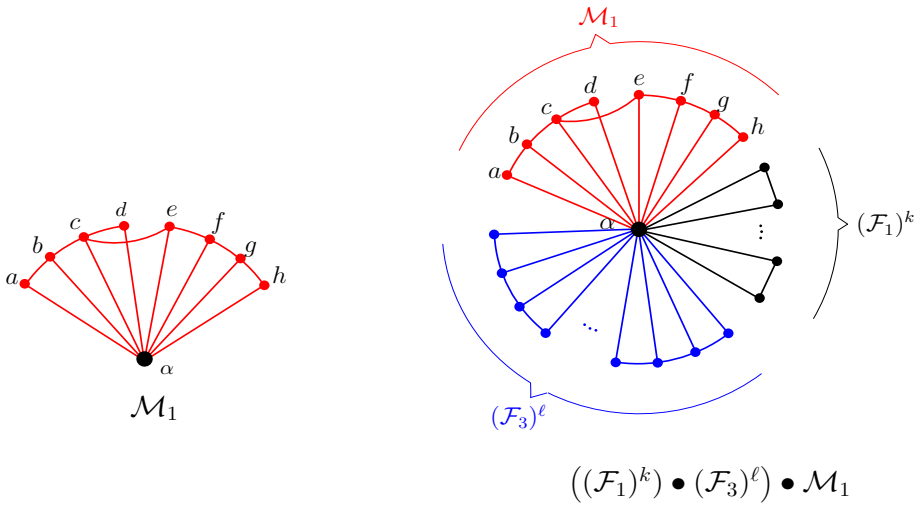


Figure B.4: $\mathcal{M}_1$ graph, and $H = ((\mathcal{F}_1)^k \bullet (\mathcal{F}_3)^\ell) \bullet \mathcal{M}_1$, as defined in the Lemma B.1.5.

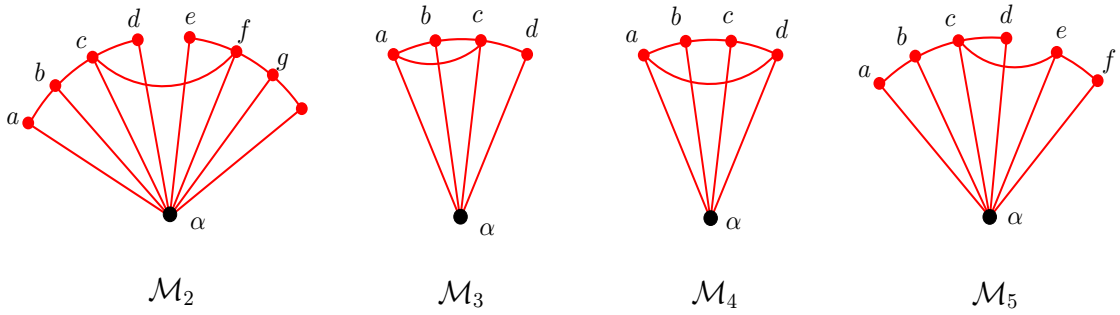


Figure B.5: $\mathcal{M}_2$, $\mathcal{M}_3$, $\mathcal{M}_4$, and $\mathcal{M}_5$ graphs.

Similarly, we can prove the following result,

Lemma B.1.7 *Let $H = \mathcal{M}_3 \bullet ((\mathcal{F}_1)^k \bullet (\mathcal{F}_3)^\ell)$, as shown in the Fig. B.6. Then,*

$$K_f(H) \;=\; \frac{2}{21}\ell(92\ell + 154.9) \;+\; \frac{2}{3}k(4k + 12.85) \;+\; \frac{204}{21}k\ell \;+\; 5.75 \qquad (99)$$

Also,

Lemma B.1.8 *Let $H = \mathcal{M}_4 \bullet ((\mathcal{F}_1)^k \bullet (\mathcal{F}_3)^\ell)$, as shown in the Fig. B.6. Then,*

$$K_f(H) \;=\; \frac{2}{21}\ell(92\ell + 151.4055) \;+\; \frac{2}{3}k(4k + 12.6004) \;+\; \frac{204}{21}k\ell \;+\; 5.333 \qquad (100)$$

Finally, we state the following result.

Lemma B.1.9 *Let $H = \mathcal{M}_5 \bullet ((\mathcal{F}_1)^k \bullet (\mathcal{F}_3)^\ell)$, as shown in the Fig. B.6. Then,*

$$K_f(H) \;=\; \frac{2}{21}\ell(92\ell + 249.6662) \;+\; \frac{2}{3}k(4k + 20.3333) \;+\; \frac{204}{21}k\ell \;+\; 15.7778 \quad (101)$$

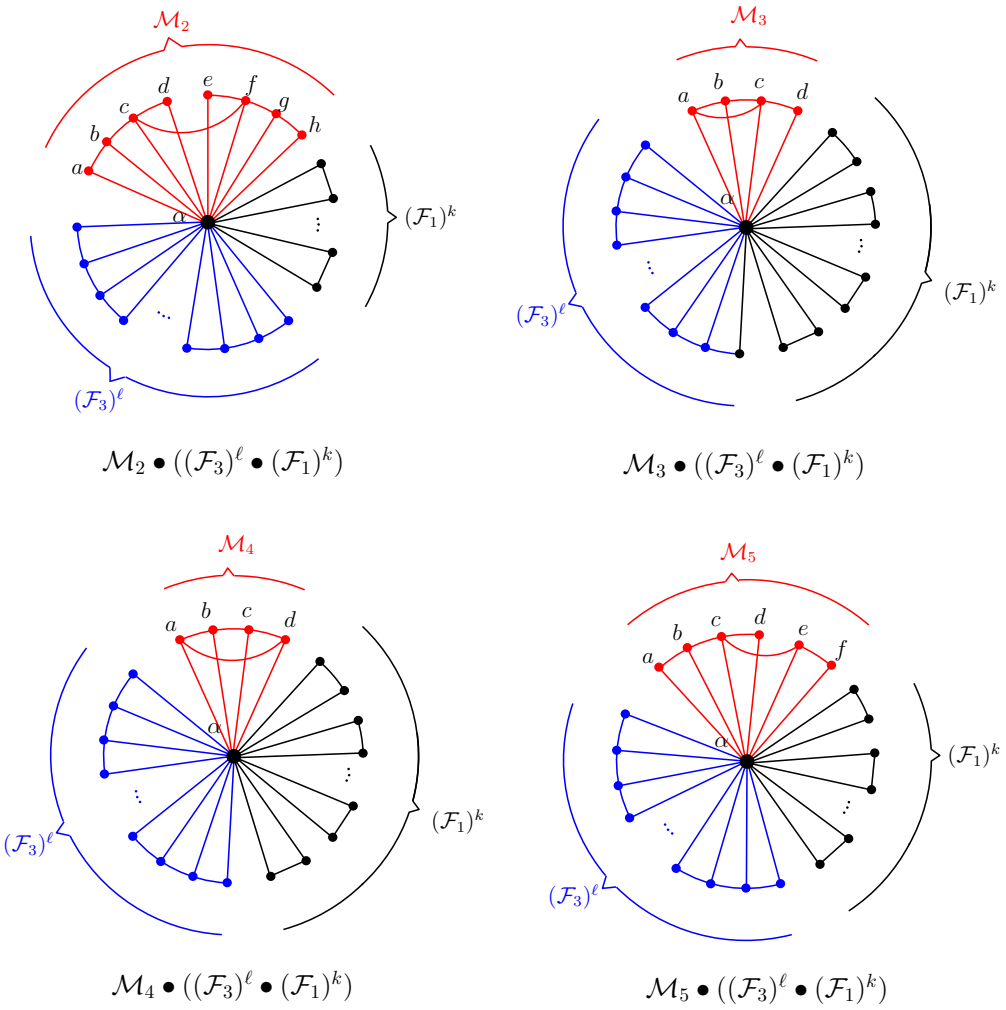


Figure B.6: Various graphs used in Lemmas B.1.6 – B.1.9

B.2 Main Theorem

Now using the Lemmas B.1.1 – B.1.9, we prove Theorem 6.2.2.

Let H be a graph obtained by adding an edge between any two non-adjacent vertices in $G = (\mathcal{F}_1)^k \bullet (\mathcal{F}_3)^\ell$. Then, H is isomorphic to one of the following graphs,

$$
\begin{aligned}
(1) \quad H_1 &= (\mathcal{F}_1)^{k-2} \bullet \mathcal{F}_{\ell+1} \\
(2) \quad H_2 &= ((\mathcal{F}_1)^{k} \bullet (\mathcal{F}_3)^{\ell-2}) \bullet \mathcal{F}_7 \\
(3) \quad H_3 &= ((\mathcal{F}_1)^{k-1} \bullet (\mathcal{F}_3)^{\ell-1}) \bullet \mathcal{F}_5 \\
(4) \quad H_4 &= ((\mathcal{F}_1)^{k} \bullet (\mathcal{F}_3)^{\ell-2}) \bullet \mathcal{M}_1 \\
(5) \quad H_5 &= ((\mathcal{F}_1)^{k} \bullet (\mathcal{F}_3)^{\ell-2}) \bullet \mathcal{M}_2 \\
(6) \quad H_6 &= ((\mathcal{F}_1)^{k} \bullet (\mathcal{F}_3)^{\ell-1}) \bullet \mathcal{M}_3 \\
(7) \quad H_7 &= ((\mathcal{F}_1)^{k} \bullet (\mathcal{F}_3)^{\ell-1}) \bullet \mathcal{M}_4 \\
(8) \quad H_8 &= ((\mathcal{F}_1)^{k-1} \bullet (\mathcal{F}_3)^{\ell-1}) \bullet \mathcal{M}_5
\end{aligned}
$$

where $\mathcal{M}_1,\ \mathcal{M}_2,\ \mathcal{M}_3,\ \mathcal{M}_4$ and $\mathcal{M}_5$ are shown in the Fig B.4 and Fig. B.5.

Now using, (86) and (85), we get,

$$K_f(H_2) - K_f(H_1) \ = \ (0.2916)\ell \ + \ (0.1461)k \ + \ 0.0735 \ > \ 0 \tag{102}$$

using, (90) and (85), we get,

$$K_f(H_3) - K_f(H_1) \ = \ (0.1431)\ell \ + \ (0.0714)k \ + \ 0.03539 \ > \ 0 \tag{103}$$

Similarly, from (94) and (85), we get,

$$K_f(H_4) - K_f(H_1) \ = \ (0.4157)\ell \ + \ (0.2079)k \ + \ 0.1036 \ > \ 0 \tag{104}$$

Also, from (98) and (85),

$$K_f(H_5) - K_f(H_1) \ = \ (0.5565)\ell \ + \ (0.2792)k \ + \ 0.1397 \ > \ 0 \tag{105}$$

(99) and (85) gives,

$$K_f(H_6) - K_f(H_1) \ = \ (0.9428)\ell \ + \ (0.4714)k \ + \ 0.23579 \ > \ 0 \tag{106}$$

Using (100) and (85), we get,

$$K_f(H_7) - K_f(H_1) \ = \ (0.6103)\ell \ + \ (0.3047)k \ + \ 0.1485 \ > \ 0 \tag{107}$$

Finally, (101) and (85) gives,

$$K_f(H_8) - K_f(H_1) \;=\; (0.2539)\ell \;+\; (0.1269)k \;+\; 0.06358 \;>\; 0 \qquad (108)$$

From (102) – (108), it can be seen that H_1 has minimum K_f among all possible graphs that can be obtained by adding a single edge to $G = (\mathcal{F}_1)^k \bullet (\mathcal{F}_3)^\ell$, thus proving the required result. ■

REFERENCES

[1] M. Yarvis, N. Kushalnagar, H. Singh, A. Rangarajan, Y. Liu, and S. Singh, "Exploiting heterogeneity in sensor networks," In *Proc.* 24^{th} *Annual Joint Conference of IEEE Computer and Communications Societies*, Miami, FL., pp. 878–890, March 2005.

[2] Y. Wang, X. Wang, D. P. Agrawal, and A. A. Minai, "Impact of heterogeneity on coverage and broadcast reachability in wireless sensor networks," In *International Conference on Computer Communications and Networks*, Arlington, VA., October 2006.

[3] J. D. Thomas and K. Sycara, "Heterogeneity, Stability and Efficiency in Distributed Systems," In *Proc. International Conference on Multiagent Systems*, Paris, pp. 293–300, July 1993.

[4] L. E. Parker, "Lifelong adaptation in heterogeneous multirobot teams: Response to continual variation in individual robot performance," *Autonomous Robots*, vol. 8, no. 3, pp. 239–267, 2000.

[5] T. Balch, "The impact of diversity on performance in multirobot foraging," In *Proc.* 3^{rd} *International Conference on Autonomous Agents*, pp. 92–99, 1999.

[6] D. Kumar, T. C. Aseri, and R. B. Patel, "EEHC: Energy efficient heterogeneous clustered scheme for wireless sensor networks," *Computer Communications*, vol. 32, no. 4, pp. 662–667, 2009.

[7] C. R. Rao, "Diversity and dissimilarity coefficients: A unified approach," *Theoretical Population Biology*, vol. 21, no. 1, pp. 24–43, 1982.

[8] D. Schleuter, M. Daufresne, F. Massol, and C. Argillier, "A users guide to functional diversity indices," *Ecological Monographs*, vol. 80, pp. 469–484, 2010.

[9] Z. Botta-Dukát, "Rao's quadratic entropy as a measure of functional diversity based on multiple traits," *Journal of Vegetation Science*, vol. 16, pp. 533–540, 2005.

[10] P. D. Allison, "Measures of Inequality," *American Sociological Review*, vol. 43, no. 16, pp. 865–880, 1978.

[11] F. A. Cowell, *Measuring Inequality*, Oxford University Press, 2011.

[12] T. Balch, "Hierarchic social entropy: An information theoretic measure of robot group diversity," *Autonomous Robots*, vol. 8, no. 3, pp. 209–238, 2000.

[13] E. Estrada, "Quantifying network heterogeneity," *Physical Review E*, vol. 82, no. 6, 2010.

[14] M. E. .J. Newman, "The structure and function of complex networks," *SIAM Review*, vol. 45, no. 2, pp. 167–256, 2003.

[15] P. Balvanera, A. B. Pfisterer, N. Buchmann, J. S. He, T. Nakashizuka, D. Raffaelli, and B. Schmid, "Quantifying the evidence for biodiversity effects on ecosystem functioning and services," *Ecology Letters*, vol. 9, pp. 1146–1156, 2006.

[16] P. Legendre and L. Legendre, *Numerical ecology*, Amsterdam: Elsevier, 1998.

[17] H. Hillebrand and B. Matthiessen, "Biodiversity in a complex world: consolidation and progress in functional biodiversity research," *Ecology Letters*, vol. 12, pp. 1405–1419, 2009.

[18] S. Rodan and C. Galunic, "More than network structure: How knowledge heterogeneity influences managerial performance and innovativeness," *Strategic Management Journal*, vol. 25, no. 6, pp. 541–562, 2004.

[19] W. Abbas and M. Egerstedt,"Hierarchical assembly of leader-asymmetric single leader networks," In *Proc. American Control Conference*, San Francisco, CA., pp. 1082–1087, 2011.

[20] W. Abbas and M. Egerstedt,"Distribution of agents in heterogeneous multiagent systems," In *Proc. IEEE Conference on Decision and Control*, Orlando, FL., pp. 976–981, 2011.

[21] W. Abbas and M. Egerstedt,"Distribution of agents with multiple capabilities in heterogeneous multiagent networks – A graph theoretic view," In *IEEE Conefernce on Decision and Control*, Maui, HI., 2012.

[22] W. Abbas and M. Egerstedt,"Securing multiagent systems against a sequence of intruder attacks," In *Proc. American Control Conference*, Montreal, Canada, 2012.

[23] W. Abbas and M. Egerstedt, "Robust graph topologies for networked systems," In *Proc. IFAC Workshop on Estimation and Control of Networked Systems*, Santa Barbara, CA., 2012.

[24] Y. Yazicioglu, W. Abbas, and M. Egerstedt, "A tight lower bound on the controllability of networks with multiple leaders," In *Proc. IEEE Conference on Decision and Control*, Maui, HI., 2012.

[25] W. Abbas and M. Egerstedt, "Energy-efficient data collection in heterogeneous wireless sensor and actor networks," In *IEEE Conefernce on Decision and Control*, Florence, Italy, 2013.

[26] W. Abbas, M. Egerstedt, C.-H. Liu, R. Thomas, and P. Whalen, "Deploying robots with two sensors in $K_{1,6}$-free graphs," (submitted) *Journal of Graph Theory*, available at arXiv:1308.5450

[27] W. Abbas and M. Egerstedt, "Heterogeneity in cooperative networks in terms of the resources distribution," (submitted) *Commuications in Information and Systems*.

[28] W. Abbas and M. Egerstedt, "On the value of heterogeneity for efficient complete coverage," (submitted) *IET Wireless Sensor Systems*.

[29] S. Diaz and M. Cabido, "Vive la difference: Plant functional diversity matters to ecosystem processes," *Trends in Ecology & Evolution*, vol. 16, pp. 646–655, 2001.

[30] C. R. Rao and T. K. Nayak, "Cross entropy, dissimilarity measures, and characterizations of Quadratic Entropy," *IEEE Transactions on Information Theory*, vol. IT-31, pp. 589–593, 1985.

[31] S. Pavoine, S. Ollier, and D. Pontier, "Measuring diversity from dissimilarities with Rao's quadratic entropy: Are any dissimilarities suitable?" *Theoretical Population Biology*, vol. 67, no. 4, pp. 231–239, 2005.

[32] L. Han, E. Hancock, and R. Wilson, "Entropy versus heterogeneity for graphs," in X. Jiang, M. Ferrer, and A. Torsello (Eds.), *Graph-Based Representations in Pattern Recognition*, vol. 6558 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 32–41, 2011.

[33] T. Snijders, "The degree variance: An index of graph heterogeneity," *Social Networks*, vol. 3, pp. 163–174, 1981.

[34] J. A. J. Hendrickson and P. R. Ehrlich, "An expanded concept of species diversity," *Notulae Naturae*, vol. 439, pp. 1–6, 1971.

[35] R. M. Warwick and K .R. Clarke, "New biodiversity measures reveal a decrease in taxonomic distinctness with increasing stress," *Marine Ecology Progress Series*, vol. 129, pp. 301–305, 1995.

[36] J. C. Gower, "A general coefficient of similarity and some of its properties," *Biometrics*, vol. 27, pp. 857–871, 1971.

[37] M. W. Pienkowski, A. R. Watkinson, G. Kerby, K. R. Clarke, and R. M. Warwick, "A taxonomic distinctness index and its statistical properties," *Journal of applied Ecology*, vol. 35, no. 4, pp. 523–531, 1998.

[38] J. Izsák and L. Papp, "A link between ecological diversity indices and measures of biodiversity," *Ecological Modelling*, vol. 130, pp. 151–156, 2000.

[39] C. Ricotta and L. szeidl, "Towards a unifying approach to diversity measures: bridging the gap between the Shannon entropy and Rao's quadratic index," *Theoretical Population Biology*, vol. 70, pp. 237–243, 2006.

[40] W. B. Sherwin, "Entropy and information approaches to genetic diversity and its expression: Genomic geography," *Entropy*, vol. 12, pp. 1765–1798, 2010.

[41] L. Iocchi and D. Nardi, "Distributed coordination in heterogeneous multirobot systems," *Autonomous Robots*, vol. 15, no. 2 pp. 155–168, 2003.

[42] M. Potter, L. Meeden, and A. Schultz, "Heterogeneity in the coevolved behaviors of mobile robots: The emergence of specialists," In *International Conference on Artificial Intelligence*, Washington, 2001.

[43] S. Bandini, S. Manzoni, and C. Simone, "Heterogeneous agents situated in heterogeneous spaces," *Applied Artificial Intelligence*, vol. 16, pp. 831–852, 2002.

[44] T. Balch, *Behavioral Diversity in Learning Robot Teams*, Ph.D. Thesis, College of Computing, Georgia Institute of Technology, 1998.

[45] T. Balch and L. E. Parker, *Robot Teams: From Diversity to Polymorphism*, Wellesley: A. K. Peters Ltd., 2002.

[46] A. Howard, L. E. Parker, and G. S. Sukhatme, "Experiments with a large heterogeneous mobile robot team: exploration, mapping, deployment and detection," *International Journal of Robotics Research*, vol. 25, pp. 431–447, 2006.

[47] T. W. Haynes, S. T. Hedetniemi, and P. J. Slater, *Domination in graphs: Advanced topics*, New York: Marcel Dekker, 1998.

[48] T. W. Haynes, S. T. Hedetniemi, and P. J. Slater, *Fundamentals of domination in graphs*, New York: Marcel Dekker, 1998.

[49] S. T. Hedetniemi and R. C. Laskar, "Bibliography on domination in graphs and some basic definitions of domination parameters," *Discrete Mathematics*, vol. 86, pp. 257–277, 1990.

[50] K. Alzoubi, P.-J. Wan, and O. Frieder, "Geometric spanners for wireless ad hoc networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 14, no. 5, pp. 408–421, 2003.

[51] S. Yang, F. Dai, M. Cardei, and J. Wu, "On multiple point coverage in wireless sensor networks," In *Proc.* 2^{nd} *IEEE International Conference on Mobile Ad Hoc and Sensor Systems*, 2005.

[52] T. Moscibroda and R. Wattenhofer, "Maximizing the lifetime of dominating sets," In *Proc.* 5^{th} *IEEE International Workshop on Algorithms for Wireless, Mobile, Ad Hoc and Sensor Networks*, 2005.

[53] S. V. Pemmaraju and I. A. Pirwani,"Energy conservation via domatic partitions," In *Proc. 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, New York, pp. 143–154, 2006.

[54] P. A. Dreyer, *Applications and Variations of Domination in Graphs*, Ph.D. dissertation, Department of Mathematics, Rutgers University, New Brunswick, NJ., October 2000.

[55] P. J. Slater, "Fault-tolerant locating-dominating sets," *Discrete Mathematics*, vol. 249, pp. 179–189, 2002.

[56] W. Goddard, S. M. Hedetnieni, and S. T. Hedetniemi, "Eternal Security in Graphs," *The Journal of Combinatorial Mathematics and Combinatorial Computing*, vol. 52, pp. 160–180, 2005.

[57] F. Zou, Z. Zhang, and W. Wu, "Latency-bounded minimum influential node selection in social networks," In *Proc. 4th International Conference on Wireless Algorithms, Systems, and Applications*, Boston, MA., 2009.

[58] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, San Francisco: Freeman, 1979.

[59] E. J. Cockayne and S. T. Hedetniemi, "Towards a theory of domination in graphs," *Networks*, vol. 7, no. 3, pp. 247–261, 1977.

[60] U. Feige, M. M. Halldòrsson, G. Kortsarz, and A. Srinivasan, "Approximating the domatic number," *SIAM Journal on Computing*, vol. 32, no.1, pp. 172–195, 2002.

[61] P. Dankelmann and N. Calkin, "The domatic number of regular graphs," *Ars Combinatoria*, vol. 73, pp. 247–255, 2004.

[62] T. L. Lu, P. H. Ho, and G. J. Chang, "The domatic number problem in interval graphs," *SIAM Journal on Discrete Mathematics*, vol. 3, no. 4, pp. 531–536, 1990.

[63] M. A. Bonuccelli, "Dominating sets and domatic number of circular arc graphs," *Discrete Applied Mathematics*, vol. 12, no. 3, pp. 203–213, 1985.

[64] S. L. Peng and M. S. Chang, "A simple linear time algorithm for the domatic partition problem on strongly chordal graphs," *Information Processing Letters*, vol. 43, no. 6, pp. 287–300, 1992.

[65] S. H. Poon, W. Yen, and C. T. Ung, "Domatic partition on several classes of graphs," *Combinatorial Optimization and Applications*, pp. 245–256, 2012.

[66] S. Fujita, M. Yamashita, and T. Kameda, "A study on r-configurations–A resource assignment problem on graphs," *SIAM Journal on Discrete Mathematics*, vol. 13, pp. 227254, 2000.

[67] R. Kreshner, "The number of circles covering a set," *American Journal of Mathematics*, vol. 61, no. 3, pp. 665–671, 1939.

[68] G.F. Tóth, and W. Kuperberg, "A survey of recent results in the theory of packing and covering," in *New Trends in Discrete and Computational Geometry*, Springer Berlin Heidelberg, pp. 251–279, 1993.

[69] I. Akyildiz and I. Kasimoglu, "Wireless sensor and actor networks: research challenges," *Ad Hoc Networks*, vol. 2, pp. 351–367, 2004.

[70] I. Akyildiz and M. Vuran, *Wireless Sensor Networks*, John Wiley & Sons, New York, 2010.

[71] D. A. Anisi and P. Ögren and X. Hu, "Cooperative minimum time surveillance with multiple ground vehicles," *IEEE Transactions on Automatic Control*, vol. 55, no. 12, pp. 2679–2691, 2010.

[72] D. A. Anisi, T. Lindskog, and P. Ögren, "Algorithms for the connectivity constrained unmanned ground vehicle surveillance problem," In *Proc. European Control Conference*, Budapest, Hungary, pp. 2990–2995, 2009.

[73] D. A. Anisi, P. Ögren, and X. Hu, "Communication constrained multi-UGV surveillance," In *Proc.* 17^{th} *IFAC World Congress*, Seoul, Korea, pp. 581–586, 2008.

[74] A. Fagiolini, G. Valenti, L. Pallottino, G. Dini, and A. Bicchi, "Decentralized intrusion detection for secure cooperative multi-agent systems," In *Proc. IEEE Conference on Decision and Control*, New Orleans, LA., pp.1553–1558, 2007.

[75] F. Pasqualetti, A. Bicchi, and F. Bullo, "Distributed intrusion detection for secure consensus computations," in *Proc. IEEE Conference on Decision and Control*, pp. 5594–5599, 2007.

[76] A. Ganguli, J. Cortes, and F. Bullo, "Distributed deployment of asynchronous guards in art galleries," In *Proc. American Control Conference*, Minneapolis, MN., pp. 1416–1421, 2006.

[77] Y. Guo, L. Parker, and R. Madhavan, "Towards collaborative robots for infrastructure security applications," In *Proc. International Symposium on Collaboration Technologies and Systems*, pp. 235–240, 2004.

[78] B. Gerkey, S. Thrun, and G. Gordon, "Visibility-based pursuit-evasion with limited field of view," *Int. J. Robotics Research*, vol. 24, no. 4, pp. 299–315, 2006.

[79] V. Isler, S. Kannan, and S. Khanna, "Randomized pursuit-evasion in a polygonal environment," *IEEE Transactions on Robotics*, vol. 21, no. 5, pp.875–884, 2005.

[80] R. Vidal, O. Shakernia, H. J. Kim, D. H. Shim, and S. Sastry, "Probabalistic pursuit-evasion games: theory, implementation and experimental evaluation," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 662–669, 2002.

[81] T. D. Parsons, "Pursuit-evasion in a graph," In *Theory and Applications of Graphs, Lecture Notes in Mathematics*, Y. Alavi, and D. Lick, Eds. Springer-Verlag, vol. 642, pp. 426–441, 1976.

[82] A. Kolling and S. Carpin, "The GRAPH-CLEAR problem: defnition, theoretical properties and its connections to multirobot aided surveillance," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1003–1008, 2007.

[83] F. V. Fomin and D. M. Thilikos, "An annotated bibliography on guaranteed graph searching," *Theoretical Computer Science*, vol. 399, pp. 236–245, 2008.

[84] A. P. Burger, E. Cockayne, W. Gründlingh, C. M. Mynhardt, J. H. van Vuren, and W. Winterbach, "Infinite order domination in graphs," *Journal of Combinatorial Mathematics and Combinatorial Comouting*, vol. 50, pp. 179–194, 2004.

[85] W. Klostermeyer and G. MacGillivray, "Eternal security in graphs of fixed independence number," *Journal of Combinatorial Mathematics and Combinatorial Computing*, vol. 63, pp. 97–101, 2007.

[86] W. Klostermeyer and G. MacGillivray, "Eternally secure sets, independence sets and cliques," *AKCE International Journal of Graphs and Combinatorics*, vol. 2, pp. 119–122, 2005.

[87] W. Klostermeyer and C. M. Mynhardt, "Vertex covers and eternal dominating sets", *Discrete Applied Mathematics*, vol. 160, pp. 1183–1190, 2012.

[88] M. Fiedler,"Algebraic connectivity of graphs," *Czechoslovak Mathematical Journal*, vol. 23, pp. 298–305, 1973.

[89] L. Freeman, "A set of mesures of centrality based on betweenness," *Sociometry*, vol. 23, pp. 35–41, 1977.

[90] W. Ellens, F. Spieksma, P. Van Mieghem, A. Jamakovic, and R. Kooij, "Effective graph resistance," *Linear Algebra Appl.*, vol. 435, pp. 2491–2506, 2011.

[91] G. Young, L. Scardovi, and N. Leonard, "Robustness of noisy consensus dynamics with directed communication," In *Proc. American Control Conference*, Baltimore, MD, pp. 6312–6317, 2010.

[92] G. Young, L. Scardovi, and N. Leonard, "Rearranging trees for robust consensus," In *Proc. IEEE Conference on Decision and Control*, Orlando, FL, pp. 1000–1005, 2011.

[93] P. Barooah and J. Hespanha,"Graph effective resistance and distributed control: Spectral properties and applications," In *Proc. IEEE Conf. Decision and Control*, San Diego, CA, pp. 3479–3485, 2006.

[94] D. Klein and M. Randić, "Resistance distance," *Journal of Mathematical Chemistry*, vol. 12, pp. 81–95, 1993.

[95] L. Xiao, S. Boyd, and S. Kim, "Distributed average consensus with least-mean-square deviation," *Journal of Parallel and Distributed Computing*, vol. 67, pp. 33–46, 2007.

[96] S. Kar and J. Moura, "Distributed consensus algorithms in sensor networks with imperfect communication: Link failures and channel noise," *IEEE Transactions on Signal Processing*, vol. 57,pp. 355–369, 2009.

[97] J. Wu, M. Barahona, Y. Tan, and H. Deng, "Spectral measure of structural robustness in complex networks," *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, vol. 6, pp. 1244–1252, 2011.

[98] X.-Y. Li, "Applications of computational geometry in wireless ad hoc networks," in *Ad Hoc Wireless Networking*, X.-Z. Cheng, X. Huang, and D.-Z. Du, Eds., Kluwer, 2003, pp. 197–264.

[99] J. Cortés, S. Martinéz, and F. Bullo, "Robust rendezvous for mobile autonomous agents via proximity graphs in arbitrary dimensions," *IEEE Transactions on Automatic Control*, vol. 51, no. 8, pp. 1289–1298, 2006.

[100] E. J. van Leeuwen, "Approximation algorithms for unit disk graphs," In *Proc. IEEE International Workshop on Graph Theoretic Algorithms*, Metz, France, pp. 351–361, 2005.

[101] C. H. Liu, P. Whalen, and R. Thomas,"$(5, 2)$-configuration on minimum degree at least two, $K_{1,6}$-free graphs," 36^{th} *Annual SIAM Southeastern Atlantic Section Conference*, Huntsville, AL., March 2012.

[102] C. Bron and J. Kerbosch, "Algorithm 457: Finding all cliques of an undirected graph," *Communications of the ACM*, vol. 16, no. 9, pp. 575–577, 1973.

[103] V. V. Vazirani, *Approximation Algorithms*, Berlin: Springer-Verlag, 2001.

[104] C. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems*, Kluwer Academic Publishers, 1999.

[105] J. Gross and J. Yellen, *Graph Theory and its Applications*, Chapman and Hall/CRC, 2005.

[106] J. Brown and R. Hoshino, "Independence polynomials of circulants with an application to music," *Discrete Mathematics*, vol. 309, pp. 2292–2304, 2009.

[107] H. Fleischner, "The square of every two-connected graph is Hamiltonian," *Journal of Combinatorial Theory, Series B*, vol. 16, pp. 29–34, 1974.

[108] D. Stoyan, W. S. Kendall, and J. Mecke, *Stochastic Geometry and its Applications*, John Wiley & Sons, 1996.

[109] P. Salvìk, "A tight analysis of the greedy algorithm for set cover," *Journal of Algorithms*, vol. 25, no. 2, pp. 237–254, 1997.

[110] F. Kuhn, *The Price of Locality: Exploring the Complexity of Distributed Coordination Primitives*, PhD thesis, ETH Zürich, 2005.

[111] B. Liang and Z. J. Haas, "Virtual backbone generation and maintenance in ad hoc network mobility management," In *Proc. IEEE Intl. Conf. Computer Communications*, Tel Aviv, Israel, 2000.

[112] P. Martin, R. Glvan-Guerra, and M. Egerstedt, "Power-Aware sensor coverage: An optimal control approach," in 19^{th} *International Symposium on Mathematical Theory of Networks and Systems*, Budapest, Hungary, 2010.

[113] H. Jaleel, A. Rahmani, and M. Egerstedt, "Probabilistic lifetime maximization of sensor networks," *IEEE Transactions on Automatic Control*, vol. 58, no. 2, pp. 534–539, 2013.

[114] M. Mesbahi and M. Egerstedt, *Graph Theoretic Methods in Multiagent Networks*, Princeton University. Press, 2010.

[115] R. Williams, *The Geometrical Foundation of Natural Structure : A Source Book of Design*, Dover Publications, New York, 1979.

[116] B.D.O. Anderson, C. Yu, B. Fidan, and J.M. Hendrickx, "Rigid graph control architectures for autonomous formations," *IEEE Control Systems Magazine*, vol. 28, no. 6, pp. 48–63, 2006.

[117] J.M. Hendrickx, B.D.O. Anderson, J.-C. Delvenne, and V.D. Blondel, "Directed graphs for the analysis of rigidity and persistence in autonomous agent systems," *International Journal of Robust Nonlinear Control*, vol. 17, pp. 960–981, 2007.

[118] J.M. Hendrickx, B. Fidan, C. Yu, B.D.O. Anderson, and V.D. Blondel, "Elementary operations for the reorganization of minimally persistent formations," in *Proc. Mathematical Theory of Networks and Systems Conf.*, Kyoto, Japan, pp. 859–873, 2006.

[119] J. Graver, *Counting on Frameworks: Mathematics to Aid the Design of Rigid Structures*, Mathematical Association of America, 2001.

[120] B. Smith, *Automatic Coordination and Deployment of Multi-robot Systems*, Ph.D Thesis, Georgia Institute of Technology, Atlanta, 2009.

[121] A. Chandra, P. Raghavan, W. Ruzzo, R. Smolensky, and P. Tiwari, "The electrical resistance of a graph captures its commute and cover times," *Comput. Compl.*, vol. 6, pp. 312–340, 1996.

[122] F. Hall, "The adjacency matrix, standard laplacian, and normalized laplacian, and some eigenvalue interlacing results," Tech. rep. Available from: http://www2.cs.cas.cz/semincm/lectures/2010-04-13-Hall.pdf

[123] B. Mohr, "Eigen values, diameter, and mean distance in graphs" *Graphs and Combinatorics*, vol. 57, pp. 53–64, 1991.

[124] E. Estrada and N. Hatano, "Resistance Distance, Information Centrality, Node Vulnerability and Vibrations in Complex Networks," *Network Science*, Springer London, pp. 13–29, 2010.

[125] A. Tizghadam and A. Leon-Garcia, "Betweenness centrality and resistance distance in communication networks," *IEEE Network*, vol. 24, pp. 10–16, 2010.

[126] J. Baras and P. Hovareshti, "Efficient and robust communication topologies for distributed decision making in networked systems," In *Proc. IEEE Conference on Decision and Control*, pp. 3751–3756, Shanghai, China, 2009.