DECENTRALIZED GRAPH PROCESSES FOR ROBUST MULTI-AGENT NETWORKS

A Dissertation
Presented to
The Academic Faculty

By

A. Yasin Yazıcıoğlu

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

School of Electrical and Computer Engineering
Georgia Institute of Technology
December 2014

DECENTRALIZED GRAPH PROCESSES FOR ROBUST MULTI-AGENT NETWORKS

Approved by:

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

Dr. Jeff S. Shamma, Advisor
Professor, School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Fumin Zhang
Professor, School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Eric Feron
Professor, School of Aerospace Engineering
Georgia Institute of Technology

Dr. Douglas Blough
Professor, School of Electrical and Computer Engineering
Georgia Institute of Technology

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

Date Approved: Jul 22, 2014

To my parents, Gülşen and Nafi; and my brother, Emre

ACKNOWLEDGMENT

First and foremost, I would like to express my deepest appreciation to my advisors, Dr. Magnus Egerstedt and Dr. Jeff Shamma. This work would not have been possible without their priceless guidance and endless support. They shared their invaluable knowledge and experience, trusted in my abilities, granted me freedom in my research, and provided constructive criticisms and insightful suggestions. I have learned a lot from them about research, teaching, and mentorship. They have been great sources of inspiration to me professionally and personally. It has been a great pleasure to work with them, and I am very fortunate and proud to have had them both as my advisors.

I am also indebted to Dr. Yücel Altunbaşak for recruiting me as a PhD student and offering me a research assistantship position during my first two years in the program. Furthermore, I would like to express my appreciation to Dr. Eric Feron, Dr. Fumin Zhang, Dr. Douglas Blough, and Dr. Anthony Yezzi for agreeing to be on my PhD committee and providing many useful comments about my thesis. Also, many thanks to my MS thesis advisor, Dr. Mustafa Ünel, for teaching me various skills that have been quite helpful during my PhD.

Furthermore, I would like to thank all my current and former colleagues in DCL and GRITS lab for many interesting discussions related to my research. I am fortunate to have worked alongside such kind and pleasant people in a friendly work environment. I have greatly enjoyed the lunches and the coffee breaks with Dr. Shamma's group, and the potlucks and the lab retreats with Dr. Egerstedt's group. I would also like to thank all my friends in Atlanta, especially the Turkish community at Georgia Tech, for so many pleasant memories we shared together.

I am extremely grateful to my family for all the support, encouragement, and love they have provided. They have always been my main source of strength while pursuing my goals in life. A special thanks goes to my fiancee, Derya, for her endless love and support

throughout my PhD and her invaluable comments about my research. I am very fortunate to have a partner, who completes me in both the personal and professional aspects of my life. Thanks should also go to my uncle, Dr. Engin Mermut, for inspiring scientific curiosity in me during my childhood. That curiosity has been one of the driving forces throughout my education. Many thanks to my grandmother, Fatma Mermut, for her unconditional love and support. I must also thank my younger brother, Emre, for his priceless companionship. Finally and most importantly, I would like to express my deepest gratitude to my parents for everything they have done for me. None of my achievements would have been possible without their steadfast encouragement and support.

TABLE OF CONTENTS

LIST OF FIGURES

SUMMARY

The objective of this thesis is to develop decentralized methods for building robust multi-agent networks through self-organization. Multi-agent networks appear in a large number of natural and engineered systems, including but not limited to, biological networks, social networks, communication systems, transportation systems, power grids, and robotic swarms. Networked systems typically consist of numerous components that interact with each other to achieve some collaborative tasks such as flocking, coverage optimization, load balancing, or distributed estimation, to name a few. Multi-agent networks are often modeled via interaction graphs, where the nodes represent the agents and the edges denote direct interactions between the corresponding agents. Interaction graphs play a significant role in the overall behavior and performance of multi-agent networks. Therefore, graph theoretic analysis of networked systems has received a considerable amount of attention within the last decade.

In many applications, network components are likely to face various functional or structural disturbances including, but not limited to, component failures, noise, or malicious attacks. Hence, a desirable network property is robustness, which is the ability to perform reasonably well even when the network is subjected to such perturbations.

In this thesis, robustness in multi-agent networks is pursued in two parts. The first part presents a decentralized graph reconfiguration scheme for formation of robust interaction graphs. Particularly, the proposed scheme transforms any interaction graph into a random regular graph, which is robust to the perturbations of their nodes/links. The second part presents a decentralized coverage control scheme for optimal protection of networks by some mobile security resources. As such, the proposed scheme drives a group of arbitrarily deployed resources to optimal locations on a network in a decentralized fashion.

CHAPTER 1

INTRODUCTION

Recently, there has been a rapidly growing interest in the analysis, design, and control of multi-agent networks. Such networks appear in numerous natural and engineered systems. Some examples include, but not limited to, financial networks, social networks, biological networks, communication systems, transportation systems, energy networks, sensor networks, and robotic swarms (e.g., [1, 2, 3, 4, 5, 6]). While the networks in different domains are usually distinct from each other in their functioning, they share a set of fundamental system attributes. In a nutshell, a multi-agent network typically involves multiple dynamic entities, possibly with individual decision making and control mechanisms, which are coupled through some local interactions. Such local interactions typically depend on the sensing, communication, and actuation capabilities of the agents. For instance, for a group of mobile robots, being within the sensing range of each other can imply some direct interaction between the corresponding robots. Typically, interacting agents have some direct influence on the dynamics of each other, which eventually propagates throughout the network.

Multi-agent networks can be represented via the corresponding interaction graphs, where the nodes correspond to the agents and the edges exist between the agents having some direct interaction. Interaction graphs play a significant role in the overall behavior and performance of multi-agent networks. System properties such as robustness, mixing time, and controllability are often analyzed through the topology of the interaction graph (e.g., [7, 8, 9, 10, 11]). Furthermore, any global behavior, which emerges from the local interactions among the agents, significantly depends on the interaction graph (e.g., [12, 13]). Therefore, graph theoretic analysis of networked systems has received a considerable amount of attention during the last decade (e.g., [14, 15]).

In this thesis, multi-agent networks are represented as interactions graphs, and decentralized graph processes are designed for achieving networks robust to external perturbations. Perturbations such as component failures, noise, or malicious attacks are inevitable in many applications. For instance, computer networks are prone to cyber-attacks, power grids face severe weather conditions, transportation networks are subject to delays and congestion, and sensor networks encounter various physical and functional challenges in hostile environments. Hence, multi-agent networks need to perform reasonably well even in the face of some structural and functional challenges. In general, reliable network operation depends on how well the network components are protected against such perturbations, and how well the the network can function even if some of its components are degraded (e.g., [16, 17, 18, 19]). Both of these aspects are addressed in this thesis to obtain robust multi-agent networks through self-organization. In particular, some decentralized methods are developed for formation of robust interaction graphs and network protection.

The remainder of this chapter is organized as follows: First, some graph theory preliminaries, which will be used extensively throughout the thesis, are provided. After that, an overview of the related literature about graph robustness and graph coverage is provided. Finally, this chapter concludes with stating the thesis contributions.

1.1 Graph Preliminaries

An undirected graph, $\mathcal{G} = (V, E)$, consists of a set of nodes, V, and a set of edges, $E \subseteq V \times V$, given by unordered pairs of nodes. A path is a sequence of nodes such that an edge exists between any two consecutive nodes in the sequence. For any two nodes, the distance between the nodes is equal to the number of edges in a shortest path between them. A graph is connected if there is a path between any pair of nodes. Any pair of nodes nodes are said to be adjacent if an edge exists between them. The set of nodes adjacent to a node, $i \in V$, is called its neighborhood, $\mathcal{N}_i$, i.e.

$$\mathcal{N}_i = \{j \mid (i, j) \in E\}. \tag{1}$$

For any node i, the number of nodes in its neighborhood is called its degree, d_i, i.e.,

$$d_i = |\mathcal{N}_i|, \tag{2}$$

where $|\mathcal{N}_i|$ denotes the cardinality of $\mathcal{N}_i$.

1.2 Graph Robustness

One way to assess graph robustness is based on centrality measures. Centrality measures identify the relative importance of nodes within a graph. In general, if a graph has a small number of nodes with centrality scores significantly larger than the others, then perturbations applied to those nodes have a stronger impact on the overall system (e.g., [20, 21, 22]). Hence, graphs with balanced centrality distributions are generally considered to be more robust to such targeted perturbations. Some widely used centrality measures are degree, betweenness, closeness, and eigenvector centralities. Detailed reviews on centrality measures and their applications can be found in [23, 24] and the references therein.

In addition to centrality measures, connectivity is one of the fundamental robustness measures in graph theory. A graph is said to be k-node (or -edge) connected if at least k nodes (or edges) should be removed to render the graph disconnected. In general, graphs with higher connectivity have higher robustness to targeted failure of its components [7, 8]. Connectivity can also be quantified via the second-smallest eigenvalue of the Laplacian, known as the algebraic connectivity or Fiedler eigenvalue [25]. A larger Fiedler eigenvalue implies a higher connectivity. Also, a related robustness measure is the Kirchhoff index [26], which is equal to the sum of the reciprocals of non-zero eigenvalues of the Laplacian scaled by the number of nodes. As such, a smaller Kirchhoff index generally indicates a higher robustness.

Alternatively, another measure of robustness is the expansion ratio (e.g., [27, 28]), which is quantified in terms of node and edge expansions. Expansion ratios are refined notions of connectivity. Edge expansion is also known as the isoperimetric number or the Cheeger constant. If the edge (or node) expansion of a graph is small, then it is possible to

disconnect a large set of nodes by removing only a small number of edges (or nodes). The isoperimetric number is also closely related to the algebraic connectivity, i.e. each of them is upper and lower bounded through the other (e.g., [29]). Graphs with high expansion rates are called expanders. Expanders are sparse yet well-connected, hence they are robust to noise and failures. A detailed overview of expanders and their numerous applications are presented in [30].

One class of expanders is Ramanujan graphs [31], which are contained within the family of regular graphs. A graph is called a m-regular graph if each node has m edges incident to itself. An m-regular graph is Ramanujan if the second largest (in absolute value) eigenvalue of its adjacency matrix is at most $2\sqrt{m-1}$. As such, the algebraic connectivity of a Ramunjan graph is at least $m - 2\sqrt{m-1}$. For $m \geq 3$, almost every m-regular graph is almost Ramanujan, i.e.almost every m-regular graph has the second largest eigenvalue of its adjacency matrix is at most $2\sqrt{m-1} + \epsilon$ for any $\epsilon > 0$ [28, 32]. Hence, for $m \geq 3$, a random m-regular graph of n nodes, i.e. a graph that is picked uniformly at random from the set of all m-regular graphs with n nodes, is an expander with a probability approaching 1 as n increases. Such a graph that is selected uniformly at random from the set of all m-regular graphs with n nodes is called a random regular graph.

In addition to relating network robustness to graph topology, another task is to find explicit methods for building robust graphs. One way to achieve this task is to construct expanders via graph operations such as zig-zag product (e.g., [33, 34]), or derandomized graph squaring [35]. Also, if $m-1$ is a prime power, then an explicit algebraic construction of m-regular Ramanujan graphs is presented in [36]. Furthermore, quasi Ramanujan graphs are obtained from a finite number of degree balancing operations on Watts-Strogatz small-word networks in [37].

Alternatively, a robust graph can also be built as a random m-regular graph for some $m \geq 3$. A detailed survey of the various models of random regular graphs as well as their properties can be found in [38] and the references therein. A random m-regular graph

with n nodes can be constructed by generating m copies for each node, picking a uniform random perfect matching on the nm copies, and connecting any two nodes if the matching contains an edge between their copies (e.g., [39, 40]). On the other hand, graph processes based on edge additions and removals may be designed to induce a Markov chain with a uniform limiting probability distribution over all the possible m-regular graphs with n nodes (e.g., [41, 42]). Also, a distributed scheme for building a class of regular multi-graphs, i.e. random $2m$-regular multi-graphs having m Hamiltonian cycles, is given in [43].

There are also studies on improving the robustness of a given graph through slight modifications to its topology. This is often achieved by rewiring a small percentage of the existing edges (e.g., [44]) or adding a small number of edges to the graph (e.g., [45]).

1.3 Graph Coverage

In many networks, one typical task is to provide some service over the network via distributed agents with limited capabilities. This service may take on different forms such as security, maintenance, pickup and delivery, or many others (e.g., [46, 47, 48]). One possible distributed way of providing such a service is to partition the network into responsibility regions such that each agent takes care of its own region. To this end, it is desirable to drive the mobile agents to an optimal configuration on the corresponding graph in a decentralized way. This goal can be formulated as a distributed coverage control problem on a graph.

Distributed coverage control is widely studied for continuous spaces (e.g., [49, 50, 51, 52, 53, 54]), whereas some extensions to discretized spaces (e.g., [55, 56, 57]) also exist in the literature. Essentially, distributed coverage control is a locational optimization problem, where the goal is to optimally locate a number of resources in a feasible domain. Locational optimization problems are widely studied in the literature, but most of the earlier solutions inherently assume centralized computations for a size-limited, static environment [51]. However, in distributed coverage control, solutions are expected to be implementable by agents having limited computation capabilities and partial information

available through local sensing and communications. As a possible solution, distributed behavior-based robotics was utilized in [49] for territorial multi-robot task division. On the other hand, a distributed method based on potential fields for mobile robots to spread out in an environment by moving away from nearby robots and obstacles was presented in [50].

Alternatively, one prevailing approach, first presented in [51], is to employ Lloyd's algorithm [58] to drive agents on a convex continuous space. As such, the agents are driven onto a local optimum, i.e. a centroidal Voronoi configuration. In a centroidal Voronoi partition, each point in the space is assigned to the nearest agent, and each agent is located at the center of mass of its own region. The resulting control law in [51] is decentralized in terms of Voronoi partitions, i.e. each agent needs to sense the density function within its Voronoi cell and to communicate with another agent only if their Voronoi cells share a boundary. Later on, this method was extended for agents with distance-limited sensing and communications [59] and limited power [60], as well as for heterogeneous agents covering non-convex regions [54]. Also, the requirement of sensing density functions was relaxed by incorporating methods from adaptive control and learning (e.g., [53]).

In some recent studies, distributed coverage control was studied on discretized spaces represented as graphs (e.g., [55, 56, 57]). In [55], agents are driven to centroidal Voronoi partition of the graph via a pairwise gossip algorithm. In [56], a centroidal Voronoi partition is achieved via asynchronous updates, where each agent moves to a node in its current cell leading to the maximum reduction in the local cost.

Alternatively, distributed coverage control on discrete spaces can be studied in a game theoretic framework (e.g., [57]). Game theoretic methods have been used to solve many cooperative control problems such as vehicle-target assignment (e.g., [61]), coverage optimization in static sensor networks (e.g., [62]), or dynamic vehicle routing (e.g. [63]). A game theoretic formulation was employed in [57] to drive mobile sensors with variable footprints to a power-aware optimal coverage.

Distributed coverage control on graphs is closely related to some combinatorial optimization problems such as p-median (e.g., [64]) or maximum coverage [65]. In the p-median problem, p facilities are located on some nodes of a graph in order to minimize a weighted sum of distances between each node and the nearest resource. Note that this cost function is analogous to the one employed in [51]. In the maximum coverage problem, given some subsets of weighted elements, the goal is to pick p of those subsets such that the total weight in their union is maximized. There is also a more generalized version of this problem, i.e. budgeted maximum coverage [66], where each subset has a cost, and the goal is to choose a maximum value combination subjected to the budget limit. Detailed reviews of the literature on some related combinatorial optimization problems can be found in [64, 67, 68] and the references therein.

Combinatorial optimization problems are generally NP-hard. Hence, exact solutions to these problems are unlikely (impossible if $P \neq NP$) to be obtained via polynomial-time algorithms. As such, exact algorithms to solve these problems can be arbitrarily slow in the worst case. Hence, fast approximation algorithms providing near-optimal solutions are preferred in most cases (e.g. [69, 70, 71]). Such approximations can be obtained through greedy algorithms (e.g. [69]), convex relaxations (e.g. [70]), or metaheuristics (e.g. [71]) including, but not limited to, ant colony optimization [72], evolutionary algorithms (e.g., [73]), iterated local search (e.g., [74]), simulated annealing (e.g., [75]), and tabu search (e.g., [76]).

1.4 Thesis Contributions

1.4.1 Formation of Robust Networks

Motivated by the expansion properties of almost every m-regular graph for $m \geq 3$, this thesis presents a decentralized graph reconfiguration scheme to obtain random regular interaction graphs. Using this scheme, multi-agent networks can achieve robust interaction graphs via self-organization. Formation of random regular graphs has been studied in many earlier

works (e.g., [38, 39, 40, 41, 42, 43]). However, majority of these studies present centralized algorithms (e.g., [38, 39, 40, 41]), whereas the distributed algorithms (e.g., [42, 43]) require some strong properties such as the initial graph already being a regular graph. Such properties may hold if the initial graph has emerged through a process strictly imposing them, whereas they may be very hard to satisfy if the initial graph is rather arbitrary. The first part of this thesis considers the following problem: Assume that a multi-agent network has an arbitrary connected interaction graph. How can the agents locally modify their interactions in a decentralized fashion such that the interaction graph is transformed into a random regular graph with a similar number of edges (sparsity) as the initial graph? This problem is illustrated in Fig. 1.

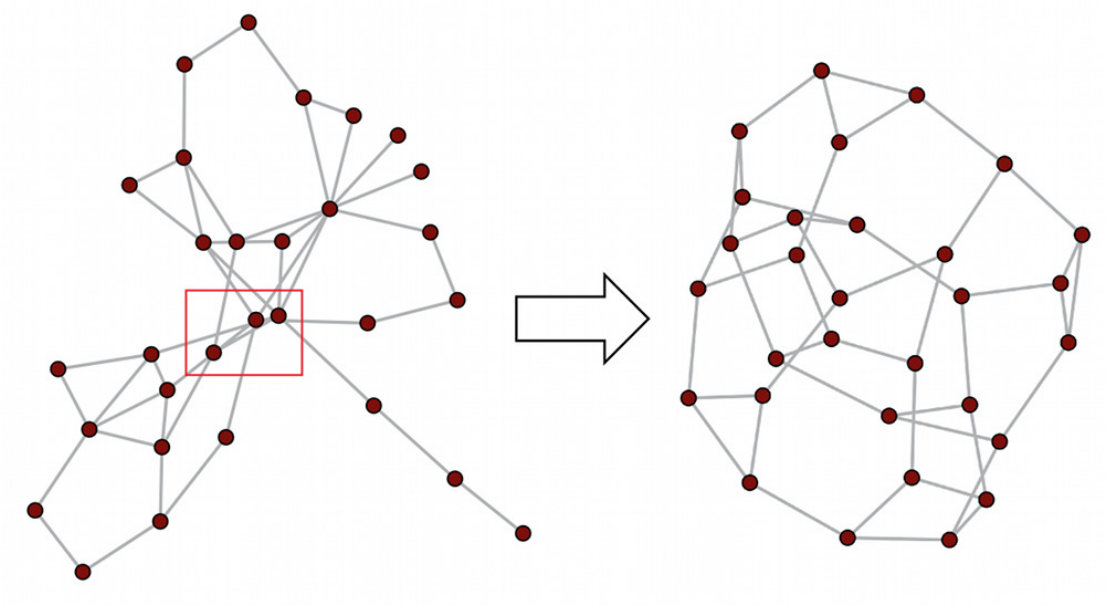


Figure 1. Agents transform a fragile interaction graph, where the failure of a few nodes (shown in the red rectangle) can disconnect half of the network, into a robust random regular graph with a similar sparsity as the initial graph.

This thesis presents a sparsity-aware decentralized scheme that asymptotically transforms any connected interaction graph into a connected random regular graph. To this end, three global objectives are simultaneously pursued while maintaining the connectivity: balance the degree distribution, randomize the local neighborhoods, and drive the average degree to an integer close to its initial value. If the average degree of the initial graph, k,

satisfies $k > 2$, then the proposed method results in a connected random m-regular graph such that $k \leq m \leq k + 2$. As such, $m \geq 3$ is ensured, and the graphs observed in the limit are (almost) Ramanujan with an arbitrarily high probability for large networks.

1.4.2 Network Protection

This thesis presents a decentralized scheme for driving a group of mobile security resources with local capabilities to an optimal configuration on any interaction graph that is unkown apriori. In particular, the second part of this thesis considers the following problem: Assume that some resources with local monitoring and protection capabilities are arbitrarily deployed on a network that is unkown apriori. How can these resources explore the network and optimize their locations in a decentralized manner to efficiently protect the system? This problem is illustrated in Fig. 2.

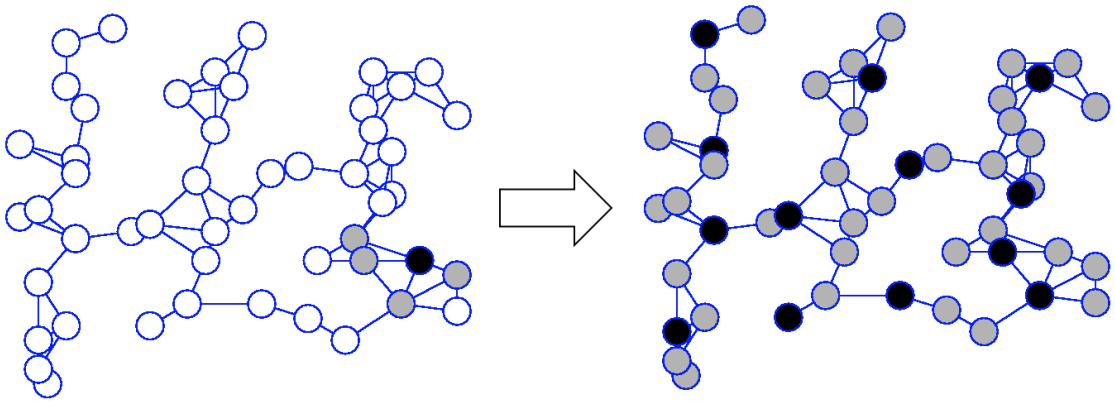


Figure 2. Some mobile security resources with local capabilities are arbitrarily deployed on an unknown network, They explore the network and optimally locate themselves to protect the network as efficiently as possible. (Black: There is at least one resource, Gray: Protected, White: Not protected)

Driving a group of mobile resources to optimal locations on a network is essentially a distributed coverage control problem on a graph. Distributed coverage control has been widely studied within the last decade for both continuous (e.g., [49, 50, 51, 52, 53, 54]) and discrete settings (e.g., [55, 56, 57]). In many of these studies (e.g., [51, 52, 53, 54, 55, 56]), mobile resources are driven to local optima by each of them taking actions locally improving its own coverage. However, when the coverage control problem is considered

on graphs with arbitrary structures, resulting local optima can be arbitrarily poor. Hence, an efficient solution needs to rely on exploration and exploitation. Such solutions can be designed by formulating the problem in a game theoretic framework (e.g. [57, 61, 62, 63]).

This thesis presents a game theoretic solution for distributed coverage control on graphs. Furthermore, different from the earlier studies on distributed coverage control, the problem is also considered for scenarios, where no explicit communications are allowed among the resources. Using the proposed scheme, a group of mobile resources can optimally protect a network by asymptotically maintaining maximum coverage with an arbitrarily high probability.

1.5 Thesis Outline

The first part of this thesis presents the proposed scheme for decentralized formation of random regular interaction graphs. The method is incrementally built in Chapters 2, 3, and 4. Chapter 2 presents a local graph transformation rule for balancing the degree distribution in a multi-agent network while maintaining the graph connectivity and the total number of edges. This scheme is extended in Chapter 3 by incorporating a neighborhood randomization rule to obtain random regular graphs in order to avoid any possible convergence to an undesired regular graph. Chapter 4 extends the proposed method to deal with the general case, where the initial average degree is not necessarily an integer.

Chapters 5 and 6 present the proposed method for distributed coverage control on graphs. In Chapter 5, the distributed graph coverage problem is solved in a game theoretic framework by designing a corresponding game and employing a learning algorithm. Chapter 6 extends the game theoretic approach presented in Chapter 5 to the cases, where no communications are allowed among the resources. In particular, a communication-free learning algorithm is designed. It is shown that maximum coverage can be asymptotically maintained with an arbitrarily high probability using the proposed algorithm. Finally, Chapter 7 concludes the thesis.

CHAPTER 2

DEGREE REGULARIZATION

This chapter presents a decentralized scheme for balancing the degree distribution in a multi-agent network. In particular, a locally applicable edge-rewiring rule is presented for this task. In order to execute the proposed rule, each node only needs to know the degrees of its immediate neighbors. The resulting dynamics preserves the graph connectivity and the total number of edges. Furthermore, any feasible trajectory is a minimizing sequence for the difference between the maximum and the minimum degree in the network (with probability 1). As such, this difference converges to 0 if a regular graph is achievable using the initial number of edges, and it converges to 1 otherwise.

2.1 Problem Formulation

For any graph $\mathcal{G}$, let $\delta(\mathcal{G})$, $\Delta(\mathcal{G})$, and $\bar{d}(\mathcal{G})$ denote the minimum, the maximum and the average degrees, respectively. A graph is said to be m-regular, if all the entries of its degree vector are equal to d, i.e. $\delta(\mathcal{G}) = \Delta(\mathcal{G}) = m$. As such, the degree non-regularity of a graph can be measured via the difference of the maximum and the minimum node degrees. Let $f(\mathcal{G})$ be the degree range defined as

$$f(\mathcal{G}) = \Delta(\mathcal{G}) - \delta(\mathcal{G}). \tag{3}$$

The objective in this chapter is to find a decentralized scheme for minimizing the degree range in a networked system while m its connectivity and the total number of edges. In multi-agent networks, connectivity is crucial for various applications since information and interactions cannot spread throughout the network without connectivity. Furthermore, each edge of an interaction graph typically implies some power consumption, communication, sensor measurement, or a physical link. As such, sparsity (having a small number of edges) is also an important feature of networked systems. In the decentralized degree regularization problem, the goal is to find some locally applicable graph transformation rules

leading to a balanced reallocation of the available edges.

Definition (Decentralized Degree Regularization (DDR) Problem): Design a locally applicable graph transformation scheme such that, for any connected $\mathcal{G}(0) = (V, E(0))$, any trajectory, $\tau = \{\mathcal{G}(0), \mathcal{G}(1), \ldots\}$, of the resulting system is a minimizing sequence for the degree range, $f(\mathcal{G}(t))$, subject to $\mathcal{G}(t)$ being connected and $|E(t)| = |E(0)|$ for every $\mathcal{G}(t) \in \tau$.

2.2 Proposed Solution

In this section, the proposed solution to the DDR problem is presented as a graph grammar, Φ_R. Before presenting Φ_R, some graph grammar preliminaries are provided below.

2.2.1 Graph Grammar Preliminaries

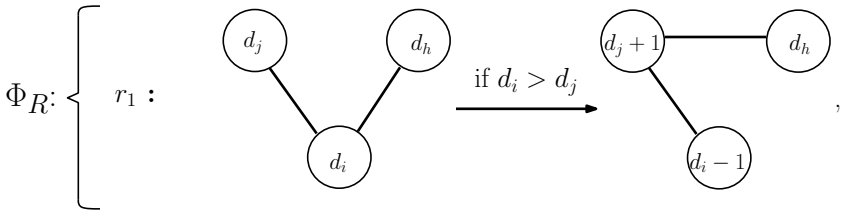
One systematic method of representing locally applicable graph transformations is to use graph grammars (e.g., [77]). A grammar, Φ, is a set of rules, where each rule is defined as a label-dependent graph transformation. More precisely, each rule is represented as an ordered pair of labeled graphs, $r = (g_l, g_r)$, where the labels represent the node states. Graph grammars operate on labeled graphs. A labeled graph, $\mathcal{G} = (V, E, l)$, consists of a node set, V, an edge set, E, and a labeling function, $l : V \mapsto \Sigma$, where Σ is the set of feasible node labels. A rule is said to be applicable to a labeled graph, $\mathcal{G} = (V, E, l)$, if $\mathcal{G}$ has a subgraph isomorphic to g_l, i.e. if there is a bijection, which preserves node labels and edges, between g_l and a subgraph of $\mathcal{G}$. A rule, $r = (g_l, g_r)$, transforms graphs isomorphic to g_l to graphs isomorphic to g_r. An initial labeled graph, $\mathcal{G}(0)$, along with a grammar, Φ, defines a non-deterministic system represented as $(\mathcal{G}(0), \Phi)$.

2.2.2 Degree Regularization

A graph grammar, $\Phi_R = \{r_1\}$, is designed as a solution to the DDR problem. In this setting, each node is labeled with its degree, i.e.

$$l(i) = d_i, \quad \forall i \in V. \tag{4}$$

The proposed grammar, Φ_R, consists of a single rule, r_1, defined as



where d_i, d_j and d_h denote the degrees of the corresponding nodes. The rule requires $d_j < d_i$, whereas d_h is arbitrary and invariant to the application of the rule.

In accordance with Φ_R, nodes behave as follows: Let i and j be two adjacent nodes, and let d_i and d_j denote their degrees, respectively. If $d_j < d_i$, then a new link is formed between j and an arbitrary neighbor of i, say h, that is not currently linked with j. At the same time, the link between i and h is terminated. As such, Φ_R only requires information available within local neighborhoods. If each node knows the degrees of its immediate neighbors, then whenever Φ_R is applicable, at least one node will be able to detect it. Note that $(\mathcal{G}(0), \Phi_R)$ is a non-deterministic dynamical system since, at any instant, a node may have multiple neighbors having degrees smaller than its own degree. Furthermore, for each such less-connected neighbor, the node may have multiple exclusive neighbors that can be rewired to its less-connected neighbor. In such cases, a feasible option is randomly chosen.

One of the important properties for distributed systems is concurrency. In graph grammars, concurrency is modeled by the commutativity of rule applications. In particular, if an application of a rule needs the output of another rule application, then these events need to happen in order. Since Φ_R consists of a single rule, it can be simultaneously executed at distinct locations on the graph.

In Lemmas 2.1 and 2.2, it is shown that any trajectory induced by Φ_R satisfies the constraints of the decentralized degree regularization problem.

Lemma 2.1 *Let $\mathcal{G}(0)$ be a connected graph, and let $\tau = \{\mathcal{G}(0), \mathcal{G}(1), \ldots\}$ be any trajectory of $(\mathcal{G}(0), \Phi_R)$. Then, $\mathcal{G}(t)$ is connected for every $\mathcal{G}(t) \in \tau$.*

Proof: Let $\mathcal{G}(t)$ and $\mathcal{G}(t+1)$ be two consecutive graphs in τ, and let $\mathcal{G}(t)$ be connected. If $\mathcal{G}(t)$ is connected, then for every node pair, $v, v' \in V$, there exists a finite simple path P from v to v'. If P does not traverse any edge rewired in the transition from $\mathcal{G}(t)$ to $\mathcal{G}(t+1)$, then P is also a valid path on $\mathcal{G}(t)$. Otherwise, let $\{i, j, h\}$ be any node triplet such that (i, h) is traversed on P, and h is rewired from i to j in the transition from $\mathcal{G}(t)$ to $\mathcal{G}(t+1)$. For each such triplet, $\{i, h\}$ (or $\{h, i\}$) in P can be replaced with $\{i, j, h\}$ (or $\{h, j, i\}$) to obtain a valid path between v and v' on $\mathcal{G}(t+1)$. Hence, if $\mathcal{G}(0)$ is connected, then every $\mathcal{G}(t) \in \tau$ is connected. ■

Lemma 2.2 *Let $\mathcal{G}(0)$ be a graph, and let $\tau = \{\mathcal{G}(0), \mathcal{G}(1), \ldots\}$ be any trajectory of $(\mathcal{G}(0), \Phi_R)$. Then, $|E(t)| = |E(0)|$ for every $\mathcal{G}(t) \in \tau$.*

Proof: Φ_R contains a single rule that preserves the number of edges in the system. Hence, it is not possible to change the number of edges in the system via Φ_R, and the number of edges remains constant along any feasible trajectory, τ. ■

Next, the equilibrium points for the dynamics induced by Φ_R are presented. These equilibrium points are the graphs such that Φ_R is not applicable anywhere on them. In particular, the connected equilibrium points are inspected since Φ_R maintains connectivity.

Lemma 2.3 *A connected graph, $\mathcal{G}$, is an equilibrium point for Φ_R if and only if $\mathcal{G}$ is a regular graph.*

Proof: $\Rightarrow$:(Contradiction) Let $\mathcal{G}$ be an non-regular graph. Since $\mathcal{G}$ is connected, if $\mathcal{G}$ is non-regular, then there exists $i, j \in V$ such that $(i, j) \in E$ and $d_i > d_j$. Note that $d_i = |\mathcal{N}_i \setminus \mathcal{N}_j| + |\mathcal{N}_i \cap \mathcal{N}_j| + 1$ and $d_j = |\mathcal{N}_j \setminus \mathcal{N}_i| + |\mathcal{N}_i \cap \mathcal{N}_j| + 1$. If $d_i > d_j$, then $|\mathcal{N}_i \setminus \mathcal{N}_j| - |\mathcal{N}_j \setminus \mathcal{N}_i| > 0$ implying $|\mathcal{N}_i \setminus \mathcal{N}_j| > 0$. Hence, for such i and j, node i always has a neighbor that is not adjacent to node j. Consequently, Φ_R is applicable to $\mathcal{G}$ and it is not an equilibrium point.

$\Leftarrow$: If $\mathcal{G}$ is a regular graph, then by definition all the nodes have the same degree. So, there

is no pair $i, j \in V$ such that $(i, j) \in E$ and $d_i > d_j$. Consequently, Φ_R is not applicable anywhere on the graph $\mathcal{G}$, and $\mathcal{G}$ is an equilibrium point. ■

Note that for a given initial configuration, $\mathcal{G}(0)$, the maintenance of connectivity and the total number of edges define a feasible set of graphs. In general, depending on $\mathcal{G}(0)$, this feasible set may not contain an equilibrium point for Φ_R. In particular, if the average degree of the initial configuration, $\bar{d}(\mathcal{G}(0))$, is not an integer, then the feasible set does not contain any regular graph. Hence, two questions are of particular interest: 1) if the feasible set includes equilibrium points, will $(\mathcal{G}(0), \Phi_R)$ converge to an equilibrium point?, 2) how does $(\mathcal{G}(0), \Phi_R)$ behave if the feasible set does not contain any equilibrium point? These two questions are addressed by inspecting the degree range along the possible trajectories of $(\mathcal{G}(0), \Phi_R)$.

Lemma 2.4 *Let $\mathcal{G}$ be a graph, and let $f(\mathcal{G})$ denote its degree range. Then, $f(\mathcal{G}) \geq 0$. Furthermore, $f(\mathcal{G}) = 0$ if and only if $\mathcal{G}$ is a regular graph.*

Proof: By definition $\Delta(G) \geq \delta(G)$, and we have $f(\mathcal{G}) = \Delta(\mathcal{G}) - \delta(\mathcal{G}) \geq 0$. Moreover, $\delta(\mathcal{G}) \leq d_i \leq \Delta(\mathcal{G})$ for all $i \in V$. Hence, if $\Delta(\mathcal{G}) - \delta(\mathcal{G}) = 0$, we obtain $d_i = \delta(\mathcal{G}) = \Delta(\mathcal{G})$ for all $i \in V$. In that case, the corresponding graph is regular. ■

Lemma 2.5 *Let $\mathcal{G}(t)$ and $\mathcal{G}(t+1)$ be two consecutive graphs on any trajectory of $(\mathcal{G}(0), \Phi_R)$. Then, their degree ranges satisfy*

$$f(\mathcal{G}(t+1)) - f(\mathcal{G}(t)) \leq 0. \tag{5}$$

Proof: The proof is based on showing that, under Φ_R, $\Delta(\mathcal{G}(t))$ is monotonically decreasing and $\delta(\mathcal{G}(t))$ is monotonically increasing. For the sake of contradiction, assume that $\Delta(\mathcal{G}(t))$ increases using Φ_R. Then, a node, j, such that $d_j = \Delta(\mathcal{G}(t))$ participates in an application of Φ_R increasing in its degree. However, that would require $d_j = \Delta(\mathcal{G}(t)) < d_i$, whereas by definition $\Delta(\mathcal{G}(t)) \geq d_i$ for any $i \in V$. Similarly, assume that $\delta(\mathcal{G}(t))$ decreases using Φ_R. Then, a node a node, i, such that $d_i = \delta(\mathcal{G}(t))$ participates in an application of Φ_R

decreasing in its degree. However, that would require $d_j < d_i = \delta(\mathcal{G})$ whereas by definition $d_j \geq \delta(\mathcal{G}(t))$ for any $j \in V$, which is again a contradiction. Consequently, $\Delta(\mathcal{G}(t))$ is monotonically decreasing and $\delta(\mathcal{G}(t))$ is monotonically increasing, i.e. $f(\mathcal{G}(t+1)) - f(\mathcal{G}(t)) \leq 0$.

■

Corollary 2.6 *Let $\tau = \{\mathcal{G}(0), \mathcal{G}(1), \ldots\}$ be any trajectory of $(\mathcal{G}(0), \Phi_R)$. The corresponding degree range sequence $\tau_f = \{f(\mathcal{G}(0)), f(\mathcal{G}(1)), \ldots\}$ converges to a finite integer, τ_f^*.*

Proof: In light of Lemma 2.4, τ_f is bounded below. Furthermore, Lemma 2.5 states that it is monotonically decreasing. Hence, one can conclude that τ_f is a convergent sequence, and since $f(\mathcal{G})$ is an integer valued function, it converges to a finite integer, τ_f^*.

■

In the remainder of this section, the almost sure convergence of τ_f to the minimum possible value that can be achieved with $|E(0)|$ edges is shown. To this end, first it is shown that for any $\mathcal{G}$, if $f(\mathcal{G}) \geq 2$, then a graph $\mathcal{G}'$ having $f(\mathcal{G}') < f(\mathcal{G})$ can be reached via Φ_R. In particular, Algorithm I provides a feasible sequence of Φ_R applications that eventually reduces the degree range as long as $f(\mathcal{G}) \geq 2$. For any such $\mathcal{G}$, let $P^* = \{i^*, j^*, \ldots, q^*\}$ denote a shortest simple path such that $d_{i^*} = \Delta(\mathcal{G})$ and $d_{q^*} = \delta(\mathcal{G})$. In other words, for any path, $P = \{i, j, \ldots, q\}$, if $d_i = \Delta(\mathcal{G})$ and $d_q = \delta(\mathcal{G})$, then let P^* satisfy $|P^*| \leq |P|$. Algorithm I transforms any $\mathcal{G}$ with $f(\mathcal{G}) \geq 2$ into a graph with a smaller degree range by recursively applying Φ_R along such P^*.

Algorithm I

1 : **input:** $\mathcal{G} = (V, E)$ s.t. $f(\mathcal{G}) \geq 2$

2 : **initialize:** $\mathcal{G}^- = \mathcal{G}, df = 0$

3 : **while** $(df = 0)$

4 : Find a $P^* = \{i^*, j^*, \ldots, q^*\}$ on $\mathcal{G}^-$,

5 : Find an $h^* \in V$ s.t. $(i^*, h^*) \in E^-$, $(j^*, h^*) \notin E^-$

6 : $E^+ = (E^- \setminus \{i^*, h^*\}) \cup \{(j^*, h^*)\}, \quad \mathcal{G}^+ = (V, E^+)$

7 : $df = f(\mathcal{G}^+) - f(\mathcal{G}^-), \quad \mathcal{G}^- = \mathcal{G}^+,$

8 : **end while**

9 : **return** $\mathcal{G}^+$

Lemma 2.7 *For any connected $\mathcal{G}$ satisfying $f(\mathcal{G}) \geq 2$, Algorithm I executes a feasible sequence of Φ_R applications resulting in a graph, $\mathcal{G}^+$, such that $f(\mathcal{G}^+) < f(\mathcal{G})$.*

Proof: For any connected graph, a shortest simple path between a node with the maximum degree and a node with the minimum degree, P^*, can always be found. Note that $d_{j^*} < d_{i^*}$ for such a $P^* = \{i^*, j^*, \ldots, q^*\}$. Since d_{i^*} is the maximum degree in the system, $d_{j^*} \leq d_{i^*}$ is immediate. Furthermore, d_{j^*} can not be equal to d_{i^*} since this would contradict with the definition of P^* by resulting a shorter path, $\{j^*, \ldots, q^*\}$, between a node with maximum degree and a node with minimum degree. Since $d_{j^*} < d_{i^*}$, a node, h^*, as described in line 5 of Algorithm I can always be found. Hence, the rewiring in line 6 is a valid application of Φ_R, and Algorithm I executes a feasible sequence of Φ_R applications.

At each iteration of the *while* loop in Algorithm I, $\mathcal{G}^+$ has either fewer nodes having a degree $\Delta(\mathcal{G})$ (i.e. $d_{j^*} < d_{i^*} - 1$), or a shorter P^* compared to $\mathcal{G}^-$ (i.e. $d_{j^*} = d_{i^*} - 1$). Note that obtaining a shorter P^* also eventually reduces the number of nodes having degree $\Delta(\mathcal{G})$. In particular, when $|P^*| = 1$, Algorithm I executes a feasible application of Φ_R, where a pair of nodes having degrees $\Delta(\mathcal{G})$ and $\delta(\mathcal{G})$ participate to have their degrees updated as $\Delta(\mathcal{G}) - 1$ and $\delta(\mathcal{G}) + 1$, respectively. As such, for any connected $\mathcal{G}$ with $f(\mathcal{G}) \geq 2$, a graph, $\mathcal{G}^+$, where each node has a degree smaller than $\Delta(\mathcal{G})$ is eventually obtained via Algorithm

I. Furthermore, since the minimum degree is monotonically increasing, $f(\mathcal{G}^+) < f(\mathcal{G})$.

■

Theorem 2.8 *Let $\mathcal{G}(0)$ be a connected graph, and let $\tau = \{\mathcal{G}(0), \mathcal{G}(1), \ldots\}$ be a feasible trajectory of $(\mathcal{G}(0), \Phi_R)$. Then, $\tau_f = \{f(\mathcal{G}(0)), f(\mathcal{G}(1)), \ldots\}$, almost surely converges to an integer, $\tau_f^* \in \{0, 1\}$.*

Proof: Corollary 2.6 shows that τ_f converges to an integer τ_f^*. If $\tau_f^* \geq 2$, then $f(\mathcal{G}(t)) \geq 2$ has to be satisfied for an infinitely long interval. However, for any connected $\mathcal{G}(t)$ satisfying $f(\mathcal{G}(t)) \geq 2$, Lemma 2.7 implies a non-zero probability that the degree range will decrease after a finite sequence of Φ_R applications. Hence, if $\mathcal{G}(0)$ is connected, then the probability that $f(\mathcal{G}(t)) \geq 2$ is satisfied for an infinitely long interval under Φ_R is 0. Consequently, $\tau_f^* \in \{0, 1\}$ with probability 1. ■

Next, the result in Theorem 2.8 is improved for the cases, where it is possible to form a regular graph, i.e. $\bar{d} \in \mathbb{N}$.

Lemma 2.9 *Let $\mathcal{G} = (V, E)$ be a non-regular graph, let $\bar{d}(\mathcal{G})$ be the average degree of $\mathcal{G}$, and let $f(\mathcal{G})$ be the degree range of $\mathcal{G}$. If $\bar{d}(\mathcal{G}) \in \mathbb{N}$, then $f(\mathcal{G}) \geq 2$.*

Proof: For any graph $\mathcal{G} = (V, E)$, its degree vector, d, can always be expressed as

$$d = \delta(\mathcal{G})\mathbf{1} + \tilde{d}, \tag{6}$$

where $\mathbf{1}$ is a vector having all its entries equal to 1. Since there is at least one node having the minimum degree, $\tilde{d}$ has at least one entry equal to 0. The average degree, $\bar{d}(\mathcal{G})$, satisfies

$$\bar{d}(\mathcal{G}) = \frac{1}{|V|}\mathbf{1}^T d = \delta(\mathcal{G}) + \frac{1}{|V|}\mathbf{1}^T \tilde{d}. \tag{7}$$

Using (7),

$$\bar{d}(\mathcal{G}) - \delta(\mathcal{G}) = \frac{1}{|V|}\mathbf{1}^T \tilde{d}. \tag{8}$$

Note that if $\bar{d}(\mathcal{G}) \in \mathbb{N}$ and $\mathcal{G}$ is non-regular, then $\bar{d}(\mathcal{G}) - \delta(\mathcal{G}) \geq 1$, and (8) implies

$$\frac{1}{|V|}\mathbf{1}^T\tilde{d} \geq 1. \tag{9}$$

Since at least one entry of $\tilde{d}$ is equal to 0, $\tilde{d}$ can have at most $|V| - 1$ positive entries. In light of (9), the sum of those positive entries are greater than or equal to $|V|$. Hence, at least one of them is greater than 1, and $f(\mathcal{G}) \geq 2$. ■

Theorem 2.10 *Let $\mathcal{G}(0)$ be a connected graph, and let $\bar{d}(\mathcal{G}(0))$ denote its average degree. If $\bar{d}(\mathcal{G}(0)) \in \mathbb{N}$, then $(\mathcal{G}(0), \Phi_R)$ almost surely converges to a $\bar{d}(\mathcal{G}(0))$-regular graph.*

Proof: Let $\tau = \{\mathcal{G}(0), \mathcal{G}(1), \ldots\}$, be a trajectory of $(\mathcal{G}(0), \Phi_R))$. Then, in light of Theorem 2.8, the sequence $\tau_f = \{f(\mathcal{G}(0)), f(\mathcal{G}(1)), \ldots\}$ almost surely converges to an integer, $\tau_f^* \in \{0, 1\}$. Furthermore, in light of Lemma 2.9, $f(\mathcal{G}) \neq 1$ for any $\mathcal{G}$ having an integer average degree, $\bar{d}(\mathcal{G}) \in \mathbb{N}$. Note that $\bar{d}(\mathcal{G}(t)) = \bar{d}(\mathcal{G}(0))$ due to Lemma 2.2. Hence, if $\bar{d}(\mathcal{G}(0)) \in \mathbb{N}$, then almost surely $\tau_f^* = 0$ and $(\mathcal{G}(0), \Phi_R)$ converges to a $\bar{d}(\mathcal{G}(0))$-regular graph. ■

Theorems 2.8 and 2.10 imply the following corollary for the convergence of $\tau_f = \{f(\mathcal{G}_0), f(\mathcal{G}_1), \ldots\}$ along the possible trajectories of $(\mathcal{G}_0, \Phi^*)$.

Corollary 2.11 *Let $\mathcal{G}(0)$ be a connected graph and let $\tau = \{\mathcal{G}(0), \mathcal{G}(1), \ldots\}$ be a feasible trajectory of $(\mathcal{G}(0), \Phi_R)$. Then, $\tau_f = \{f(\mathcal{G}(0)), f(\mathcal{G}(1)), \ldots\}$, almost surely converges to an integer, τ_f^*, such that*

$$\tau_f^* = \begin{cases} 0 & \text{if } \bar{d}(\mathcal{G}(0)) \in \mathbb{N} \\ 1 & \text{otherwise,} \end{cases} \tag{10}$$

where $\bar{d}(\mathcal{G}(0))$ is the average degree of $\mathcal{G}(0)$.

2.3 Simulation Results

In this part, some simulation results are presented to demonstrate the proposed scheme. In the first simulation, a connected graph, $\mathcal{G}(0)$, is randomly generated using 20 nodes

and 30 edges. Note that 20 nodes and 30 edges result in $\bar{d}(\mathcal{G}(0)) = 3$. Starting from the initial configuration, nodes concurrently update their neighborhoods according to Φ_R, and the system converges to a 3-regular configuration. Some graph configurations along an arbitrary trajectory of the system $(\mathcal{G}(0), \Phi_R)$ are depicted in Fig. 3, whereas the degree range, $f(\mathcal{G}(t))$, along the same trajectory is shown in Fig. 4.

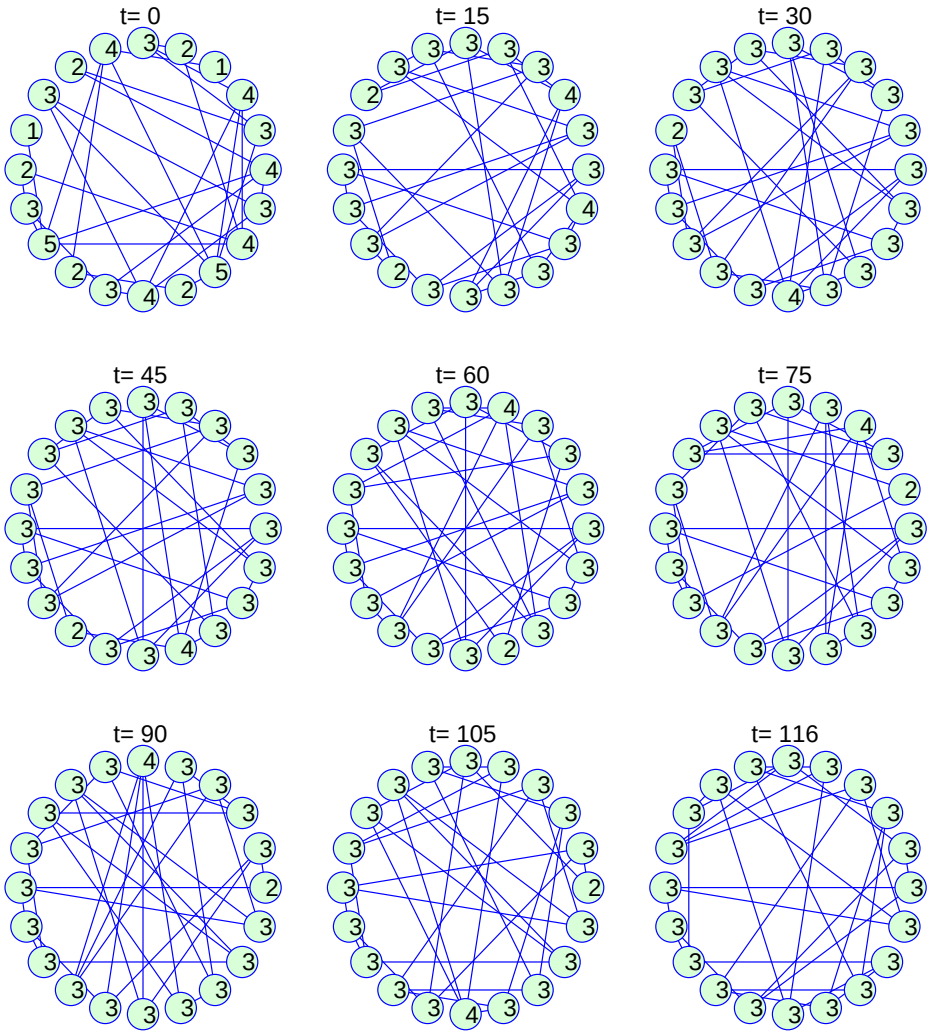


Figure 3. $\mathcal{G}(t)$ at some instants along an arbitrary trajectory of $(\mathcal{G}(0), \Phi_R)$. On this trajectory, the initial graph converges to a 3-regular graph via some concurrent applications of Φ_R in 116 time steps. On each $\mathcal{G}(t)$, nodes are labeled with their degrees.

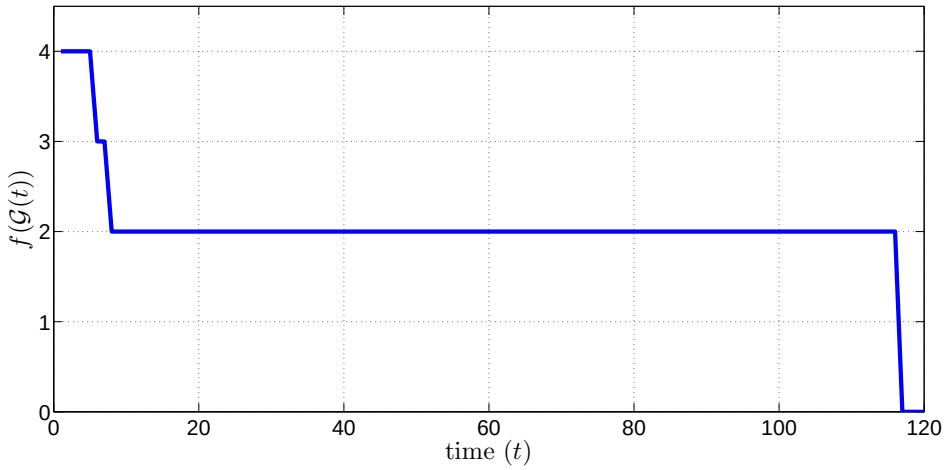


Figure 4. The degree range, $f(\mathcal{G}(t))$, along the trajectory depicted in Fig. 3.

The second simulation presents a case in which a regular configuration is not achievable. A connected initial graph is randomly generated using 20 nodes and 19 edges. As such, the initial graph is a tree. Since the average degree and the connectivity are preserved, this initial configuration almost surely converges to a tree with a degree range of 1, i.e. a path graph. Once a path graph is reached, then that structure is preserved since only the terminal nodes keep switching locations with the internal nodes adjacent to them. In the simulation, the initial graph follows a trajectory as depicted in Fig. 5, along which the degree range evolves as shown in Fig. 6. On this trajectory, the system reaches a path formation in 61 time steps.

Finally, a set of simulation results is presented to provide some insight into the expansion properties of the regular graphs obtained through Φ_R. In these simulations, integer average degrees ranging from 3 to 8 are considered. For each value of the average degree, 1000 simulations are performed. Each simulation starts with an arbitrary connected initial graph having 100 nodes. The initial graph is evolved according to Φ_R, and it converges to a regular graph. The distributions of the spectral gap for the resulting regular graphs are depicted in Fig. 7. In Fig. 7, for each value of the average degree, the smallest value of spectral gap to be defined as a Ramanujan graph is marked with a dashed vertical line.

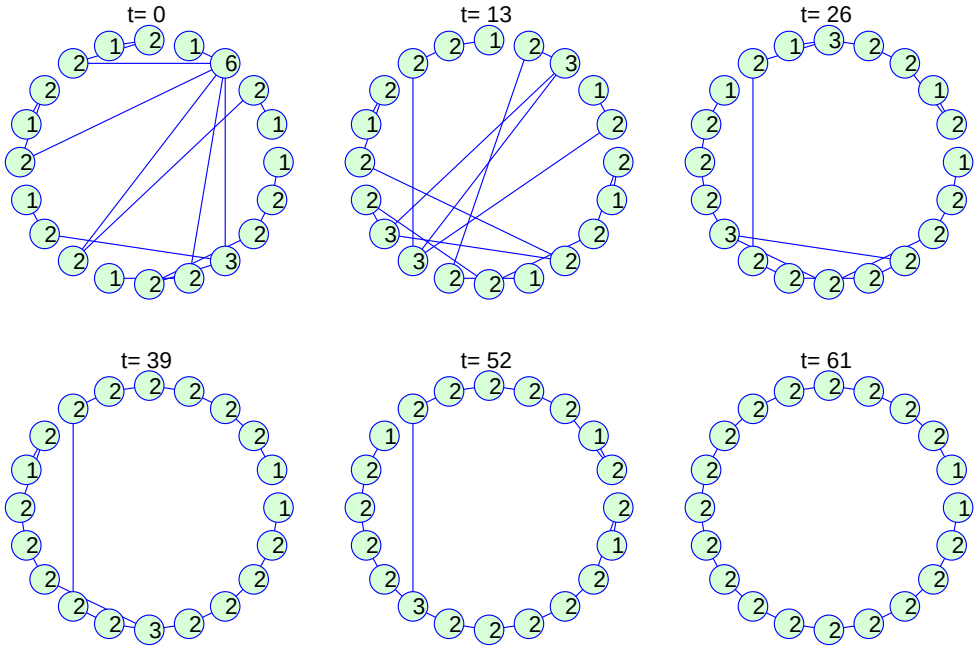


Figure 5. $\mathcal{G}(t)$ **at some instants along an arbitrary trajectory of** $(\mathcal{G}(0), \Phi_R)$**. On this trajectory, the initial graph (t=0) is formed with 20 nodes and 19 edges, and it is a tree. The initial graph reaches a path formation via some concurrent applications of** Φ_R **in 61 time steps. After this point, the path formation is preserved while the terminal nodes keep their switching locations with their immediate neighbors. On each** $\mathcal{G}(t)$**, nodes are labeled with their degrees.**

Note that while $(\mathcal{G}(0), \Phi_R)$ almost surely converges to a regular graph if $\bar{d}(\mathcal{G}(0)) \in \mathbb{N}$, the resulting graph may not be an expander. This is because the undesired rare configurations can also be attractive under Φ_R, and the probability of reaching such equilibrium points may not be converging to zero although they form a zero-measure subset of all the equilibrium points as the network size increases. However, the distributions shown in Fig. 7 suggest that if the initial configuration, $\mathcal{G}(0)$, is random, and a regular graph is reachable, then $(\mathcal{G}(0), \Phi_R)$ converges to an almost Ramanujan graph with a very high probability. As the ratio of the average degree to the network size decreases, the distribution is more concentrated around $m - 2\sqrt{m-1}$, which is the largest spectral gap a m-regular graph can have for a fixed m as the graph size goes to infinity.

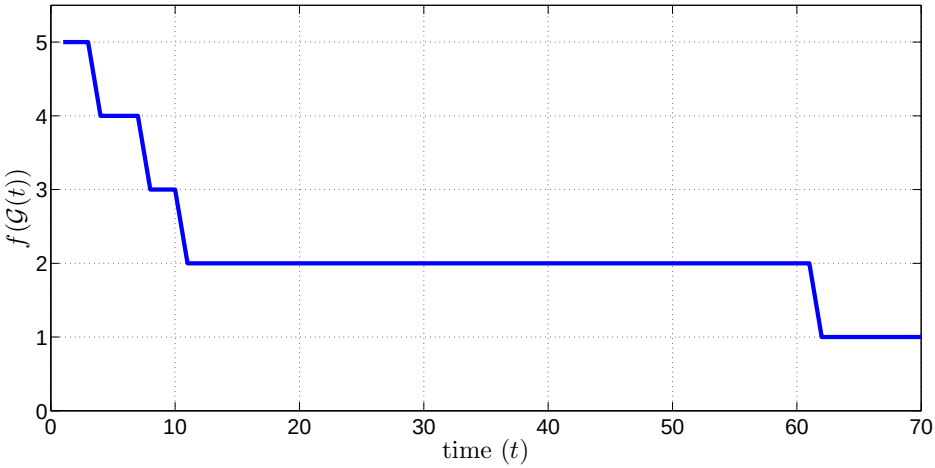


Figure 6. Degree range, $f(\mathcal{G}(t))$, along the trajectory depicted in Fig. 5.

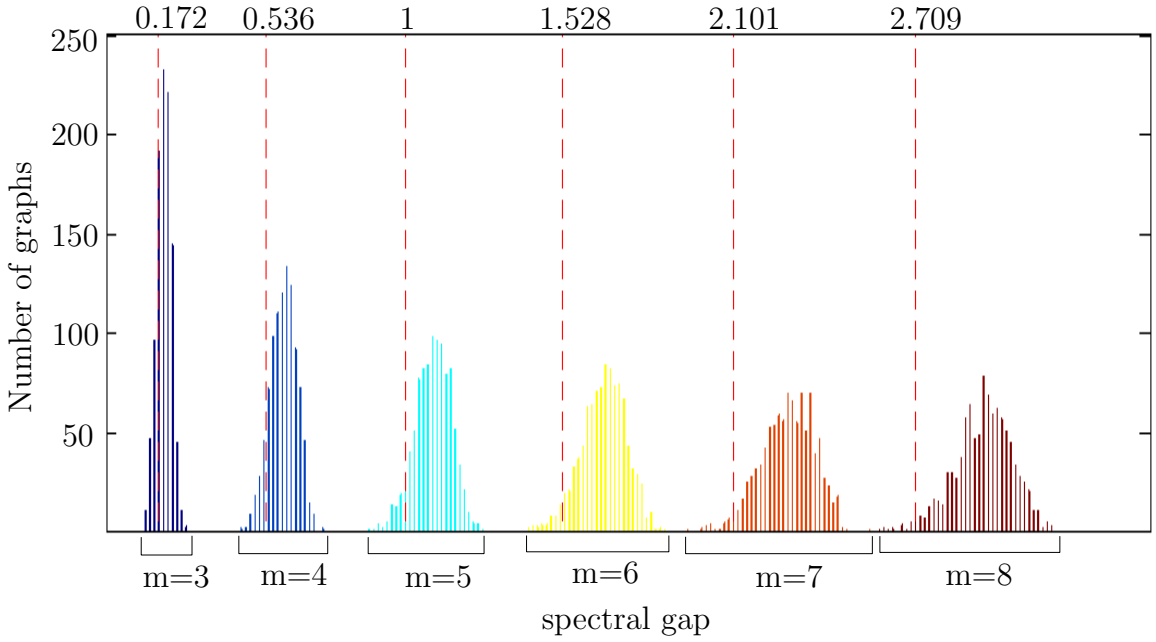


Figure 7. Spectral gap distribution for the final graphs obtained via Φ_R starting from random initial networks with 100 nodes and the average degrees of 3, 4, 5, 6, 7, and 8. For each value of the average degree, 1000 simulations are performed, each starting with a random $\mathcal{G}(0)$. The lower bound on the spectral gap to be a Ramanujan graph ($m - 2\sqrt{m-1}$) is marked as a vertical dashed line for each case.

CHAPTER 3

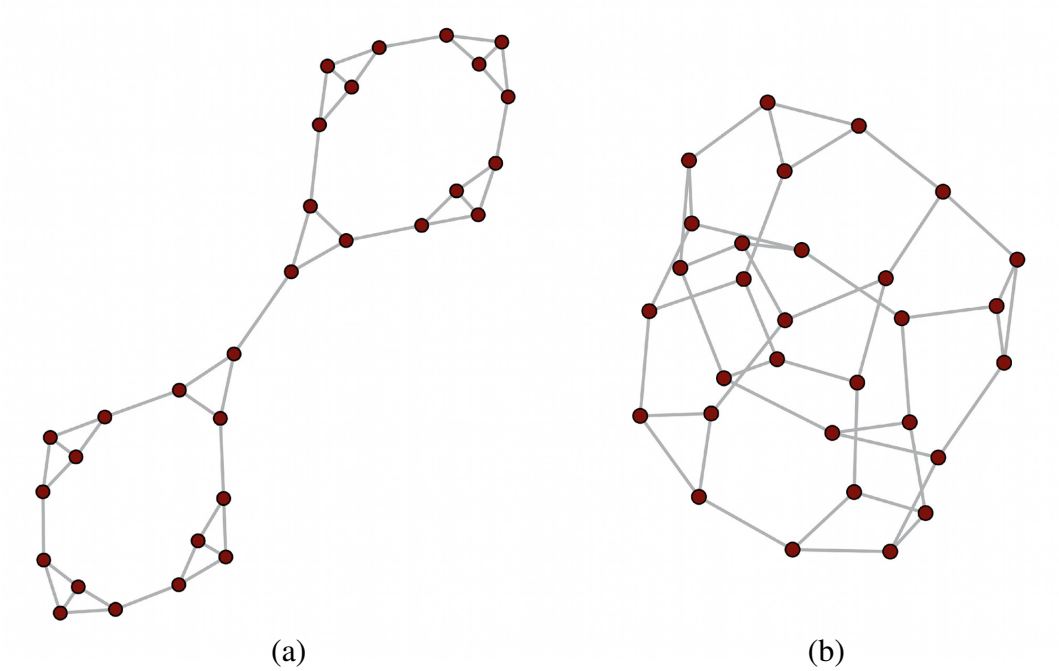
RANDOMIZED DEGREE REGULARIZATION

In this chapter, the degree regularization grammar, Φ_R, is extended to obtain a decentralized scheme leading to connected random $\bar{d}(\mathcal{G}(0))$-regular graphs, where $\bar{d}(\mathcal{G}(0))$ is the average degree of the initial graph. The proposed scheme can be represented as a graph grammar, $\Phi_{RR} = \{r_1, r_2\}$, which consists of the degree regularization rule, r_1, and a randomization rule, r_2, that was first introduced in [42]. The resulting dynamics minimize the degree differences and randomize the local neighborhoods simultaneously while maintaining the graph connectivity and the total number of edges. As such, it transforms any connected interaction graph with an integer average degree into a connected random regular graph. A distributed implementation, which leads to a uniform limiting distribution over the connected $\bar{d}(\mathcal{G}(0))$-regular graphs, is also provided.

3.1 Problem Formulation

In the previous chapter, the decentralized degree regularization (DDR) problem was considered. The DDR problem was motivated by the expansion properties of random regular graphs, and a single-rule grammar, Φ_R, was designed as a solution to the DDR problem. In Theorem 2.10, it was shown that $(\mathcal{G}(0), \Phi_R)$ almost surely converges to a $\bar{d}(\mathcal{G}(0))$-regular graph for any connected $\mathcal{G}(0)$ such that $\bar{d}(\mathcal{G}(0)) \in \mathbb{N}$. Such an initial graph, $\mathcal{G}(0)$, may converge to any of the regular graphs that is reachable via Φ_R since $(\mathcal{G}(0), \Phi_R)$ is a non-deterministic process. However, one cannot reach a conclusion about the expansion properties of the resulting graph without some information about the set of regular graphs reachable from $\mathcal{G}(0)$ via Φ_R, and the probability distribution induced by Φ_R over that reachable set. Although almost every m-regular graph is almost Ramanujan for $m \geq 3$, Φ_R can still lead to a configuration with an arbitrarily small expansion rate, with some probability depending on the initial graph. Fig. 8 illustrates a pair of 3-regular graphs on 30 nodes.

Note that both graphs are stationary under Φ_R. However, the graph in Fig. 8a is a poorly-connected configuration that has the lowest possible edge expansion ratio, i.e. have half of the network can be disconnected from the rest due the removal of a single edge.



(a) (b)

Figure 8. A poorly-connected 3-regular graph (a) and a robust 3-regular graph (b).

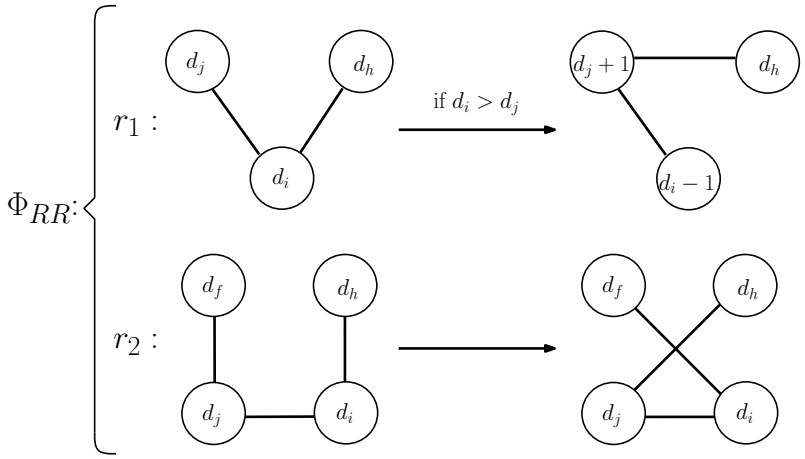
The problem addressed in this chapter is to design a locally applicable graph transformation scheme that leads to connected random regular graphs to avoid any possible convergence to poorly-connected regular graphs.

3.2 Proposed Solution

One way of avoiding any possible convergence to a poorly-connected regular graph is to modify Φ_R so that it leads to a limiting distribution over all the feasible connected regular graphs, rather than converging to one of the reachable regular graphs. In order to that, regular graphs should not be stationary under the modified grammar, and some neighborhood randomization should be active at all times without distorting the degree distribution in the network. The solution presented in this section is a modified grammar, Φ_{RR}, that achieves this task.

3.2.1 Randomized Degree Regularization

The proposed grammar, $\Phi_{RR} = \{r_1.r_2\}$, extends Φ_R by incorporating a locally applicable randomization rule (r_2), which was introduced in [42]. Under the resulting dynamics, a connected interaction graph is never stationary (unless it is a complete graph). Furthermore, it is transformed into a random regular graph as time goes to infinity, if its average degree is an integer. In Φ_{RR}, each node is labeled with its degree, and the rules are given as



In accordance with Φ_{RR}, if a node has more links than one of its neighbors, then it rewires one of its other neighbors to the less-connected node (r_1). Furthermore, adjacent nodes also exchange their exclusive neighbors (r_2). As such, while the first rule r_1 aims to balance the degree distribution, r_2 randomizes the neighborhoods.

In the remainder of this section, the dynamics induced by Φ_{RR} is analyzed. First, it is shown that Φ_{RR} maintains the graph connectivity and the average degree, and the degree range is monotonically decreasing under Φ_{RR}

Lemma 3.1 *Graph connectivity and the average degree are maintained under* Φ_{RR}.

Proof: Both r_1 and r_2 preserve the number edges, so the number of edges and the average degree are invariant to the applications of Φ_{RR}. Furthermore, both rules preserve the connectivity of the local structures. Hence, the global connectivity is also maintained under the concurrent applications on disjoint subgraphs. ■

Lemma 3.2 *The degree range, $f(\mathcal{G})$, monotonically decreases under Φ_{RR}.*

Proof: Node degrees are invariant to the applications of r_2, and the degree range can only change due to the applications of r_1. Due to Lemma 2.5, the degree range monotonically decreases under the applications of r_1. Hence, the degree range is monotonically decreasing under Φ_{RR}. ■

In general, there may be many feasible applications of Φ_{RR} on an interaction graph. In such cases, the agents need to randomly execute Φ_{RR} on disjoint subgraphs without any global coordination. To this end, Algorithm II is proposed as a distributed implementation of Φ_{RR}. By following Algorithm II , the nodes can concurrently modify their local neighborhoods in accordance with Φ_{RR} such that any feasible transformation occurs with a non-zero probability.

In accordance with Algorithm II, the nodes behave as follows: At each iteration, each node is inactive with a small probability ϵ. The inactivation probability, ϵ, ensures that any feasible application of Φ_{RR} can be realized through Algorithm II, as it will be shown in Lemma 3.3. Inactive nodes do not participate in any rule execution in that time step. Each active agent, i, picks one of its neighbors, $j \in \mathcal{N}_i$, uniformly at random, and it communicates its degree to that neighbor. Based on these communications, each active agent, i, checks the list of neighbors that picked itself, $R_i \subseteq \mathcal{N}_i$, to see if $j \in R_i$. If that is not the case, then i is a follower in that time step, i.e. it will not initiate a rule execution but it will participate if j wants to rewire i to some other node. If $j \in R_i$, then i and j are matched. Each matched pair picks a candidate rule, $r \in \Phi_{RR}$, that they will potentially execute. In Algorithm II, the agent with the larger node ID, i.e. $\max\{i, j\}$, picks the candidate rule uniformly at random. If r_1 is picked, and one of the nodes, say i, has higher degree than the other, and i is picked by at least one other neighbor ($|R_i| \geq 2$), then i chooses a neighbor, $h \neq j \in R_i$, uniformly at random. If $h \notin \mathcal{N}_j$, then r_1 is executed by rewiring h to j. If r_2 is to be executed and $|R_i|, |R_j| \geq 2$, both i and j choose one neighbor, $h \neq j \in R_i$ and $f \neq i \in R_j$, uniformly at random. If neither h nor f is linked to both i and j, then r_2 is

executed by rewiring h to j and f to i. A feasible iteration of the algorithm on a small network is illustrated in Fig. 9.

Algorithm II: Distributed Implementation of Φ_{RR}

initialize: $\mathcal{G} = (V, E)$ connected, $\epsilon \in (0, 1)$ small
repeat
  Each agent, i, is active with probability $1 - \epsilon$.
  Each active i picks a random $j \in \mathcal{N}_i$.
  For each i, $R_i = \{i' \in \mathcal{N}_i \mid i' \text{ picked } i\}$.
  for (each (i, j) s.t. $i \in R_j$, $j \in R_i$, $d_i \geq d_j$)
   $\max\{i, j\}$ picks a random $r \in \Phi_{RR}$.
    if ($r = r_1$, $d_i > d_j$, $|R_i| \geq 2$)
      i picks a random $h \in R_i \setminus \{j\}$.
      if $((j, h) \notin E)$
        $E = (E \setminus \{(i, h)\}) \cup \{(j, h)\}$.
      end if
    else if ($r = r_2$, $|R_i| \geq 2$, $|R_j| \geq 2$)
      i picks a random $h \in R_i \setminus \{j\}$.
      j picks a random $f \in R_j \setminus \{i\}$.
      if $((i, f) \notin E, (j, h) \notin E)$
        $E = (E \setminus \{(i, h), (j, f)\}) \cup \{(i, f), (j, h)\}$.
      end if
    end if
  end for
end repeat

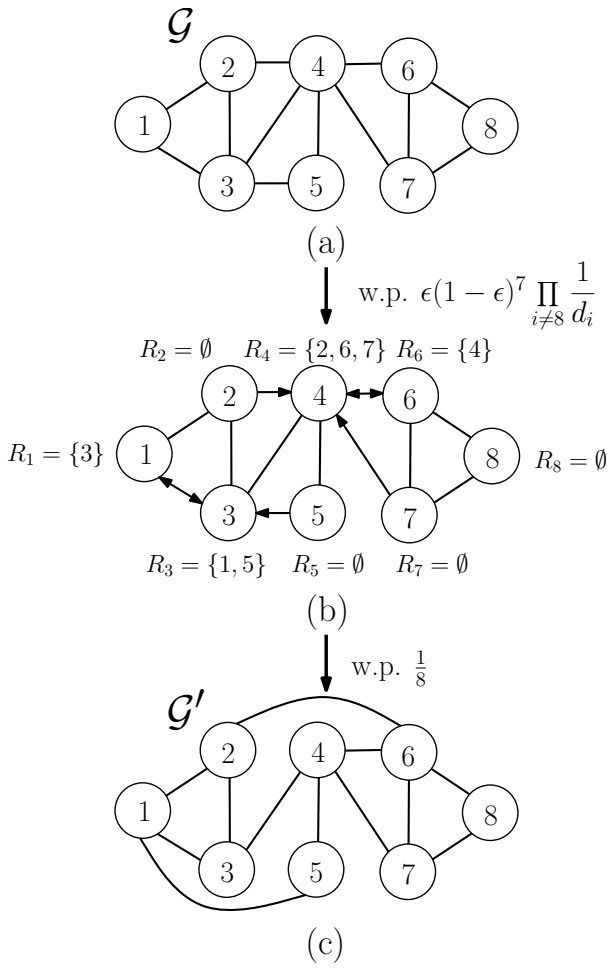


Figure 9. A feasible iteration of Algorithm II on $\mathcal{G}$ in (a) resulting in $\mathcal{G}'$ in (c) along with the probabilities of the corresponding random events. In this example, each node other than 8 is active and picks a neighbor as illustrated in (b), where each arrow is pointed from a node to its chosen neighbor. Accordingly, (1,3) and (4,6) are the matched pairs satisfying $d_3 > d_1$ and $d_4 > d_6$. With probability 0.25, nodes 3 and 6 both pick r_1 as the candidate rule. Furthermore, since $R_3 \setminus \{1\} = \{5\}$ and $R_4 \setminus \{6\} = \{2, 7\}$, 3 picks 5 to rewire with probability 0.5, and 4 picks 2 to rewire with probability 1. Hence, given the configuration in (b), $\mathcal{G}'$ can emerge with a probability of 0.125.

In the remainder of this section, the Markov chain induced by Algorithm II will be analyzed. In particular, it will be shown that if $\bar{d}(\mathcal{G}(0)) \in \mathbb{N}$, then P_{RR} has a unique limiting distribution that is uniform over the connected $\bar{d}(\mathcal{G}(0))$-regular graphs. Prior to this analysis, some background on Markov chains is provided.

3.2.2 Markov Chain Preliminaries

A Markov chain is a sequence of random variables, $X_0, X_1, X_2 \ldots$, with the Markov property, i.e.

$$\Pr[X_{t+1} = x \mid X_0 = x(0), \ldots, X_t = x(t)] = \Pr[X_{t+1} = x \mid X_t = x(t)]. \tag{11}$$

The possible values of X_t form a countable state space of the chain, $\mathcal{X}$. A Markov chain is said to be time-homogeneous if the the probability of transition between any two states is identical at each step. A time-homogeneous Markov chain is characterized by a state space, an initial state (or an initial probability distribution over the states), and a transition matrix, P, describing the state transition probabilities. The transition matrix defines the dynamics of the probability distribution over the state space, μ, as

$$\mu^T(t+1) = \mu^T(t)P. \tag{12}$$

A state x_j is said to be reachable from a state x_i if a system started in state x_i has a non-zero probability of transitioning into state x_j after a sufficient number of steps, i.e. there exists an integer, $k \geq 0$ such that

$$\Pr[X_k = x_j \mid X_0 = x_i] = P^k(x_i, x_j) > 0. \tag{13}$$

A state x_i is said to communicate with state x_j if each of them is reachable from the other. A set of states, $C \subseteq \mathcal{X}$, is a communicating class if every pair of states in C communicate with each other, and no state in C communicates with any state outside C. A communicating class, C, is closed if the probability of leaving the class is zero, i.e. any state outside C is not reachable from any state in C. A Markov chain with a finite state space always has at least one closed communicating class [78]. A Markov chain is said to be irreducible if its state space is a single communicating class.

A state, x_i, is said to be transient if, given that the chain starts x_i, there is a non-zero probability that it will never return to x_i. A state is recurrent if it is not transient. Recurrence is a class property, i.e. the states in a communicating class are either all recurrent or all

transient. If the state space is finite, then a communicating class is recurrent if and only if it is closed [78].

For any state $x_i \in \mathcal{X}$, the period of the state is defined as

$$k = \gcd\{n : \Pr[X_n = x_i \mid X_0 = x_i] > 0\}, \tag{14}$$

where 'gcd' stands for the greatest common divisor. If $k = 1$, then the state is aperiodic. As such, any x_i is aperiodic if $P(x_i, x_i) > 0$. All the states in a communicating class have the same period.

For a Markov chain with the transition matrix P, a vector, μ^*, is called a stationary distribution of the chain if it satisfies

$$\mu^{*T} = \mu^{*T} P. \tag{15}$$

Every Markov chain with a finite state space has a unique stationary distribution if and only if it has exactly one closed communicating class [79]. Hence, an irreducible Markov chain with a finite state space has a unique stationary distribution, μ^*. Furthermore, if the chain is also aperiodic, then μ^* is called the limiting distribution, and it satisfies

$$\lim_{t \to \infty} \mu(t) = \mu^*. \tag{16}$$

Moreover, if a Markov chain has a doubly stochastic state transition matrix, i.e. if the sum of elements in each column of the transition matrix is equal to 1, then the chain has a stationary distribution that is uniform over the state space. Hence, if a Markov chain with a finite state space is irreducible, aperiodic, and it has doubly stochastic transition matrix, then the its limiting distribution is uniform over the state space.

If a Markov chain is reducible, then its limiting behavior depends on the initial state as well as the limiting behavior of the chain within each closed communicating class. If a chain has multiple closed communicating classes, $\{C_1, C_2, \ldots, C_K\}$, then each closed communicating class, C_k, has its own unique stationary distribution. Extending any of these distributions to the overall chain and setting all values to zero outside the corresponding

communicating class, C_k, yields to a stationary distribution of the original chain, μ_k^*. The set of stationary distributions of the original chain is the set of all convex combinations of the extended stationary distributions, i.e. $span\{\mu_1^*, \mu_2^*, \ldots, \mu_K^*\}$. Furthermore, if each closed communicating class is aperiodic, then the chain initialized at any $x_i \in \mathcal{X}$ converges to a limiting distribution

$$\mu_{x_i}^* = \sum_{k=1}^{K} \Pr[x_i \to C_k]\mu_k^*, \tag{17}$$

where $\Pr[x_i \to C_k]$ is the probability that the chain starting at x_i eventually enters the closed communicating class C_k [80].

3.2.3 Limiting Behavior

Algorithm II is memoryless since each iteration only depends on the current graph, and the probability of any feasible transition is independent of the past. Furthermore, the average degree and the connectivity are preserved due to Lemma 3.1. As such, Algorithm II induces a Markov chain over the state space of simple connected graphs having the same number of nodes and the same average degree, i.e.

$$\mathbb{G}_{n,m} = \{\mathcal{G} = (V, E) \mid |V| = n, \bar{d}(\mathcal{G}) = m\}. \tag{18}$$

Let $\mu(t)$ denote the probability distribution over the state space at time t. Then, $\mu(t)$ satisfies

$$\mu^T(t+1) = \mu^T(t)P_{RR}, \tag{19}$$

where P_{RR} is the corresponding probability transition matrix. Accordingly, the probability of transition from any $\mathcal{G}$ to any $\mathcal{G}'$ is denoted by $P_{RR}(\mathcal{G}, \mathcal{G}')$. Note that for any pair of graphs, $\mathcal{G}, \mathcal{G}' \in \mathbb{G}_{n,m}$, the corresponding probability of transition under Algorithm II, $P_{RR}(\mathcal{G}, \mathcal{G}')$, is non-zero if and only if it is possible to transform $\mathcal{G}$ into $\mathcal{G}'$ in a single time step via Φ_{RR}.

Lemma 3.3 *Let $\mathcal{G}, \mathcal{G}' \in \mathbb{G}_{n,m}$ be any pair of graphs. $P_{RR}(\mathcal{G}, \mathcal{G}') > 0$ if and only if $\mathcal{G}'$ can be reached from $\mathcal{G}$ in one step via Φ_{RR}.*

Proof: $\Rightarrow$: Since any possible transformation in Algorithm II (lines 11, 17) satisfies a rule in Φ_{RR}, $P_{RR}(\mathcal{G}, \mathcal{G}') > 0$ implies that $\mathcal{G}'$ can be reached from $\mathcal{G}$ in one step via Φ^*.

$\Leftarrow$: Let $\mathcal{G}'$ be reachable from $\mathcal{G}$ in one step via Φ^*, and let $G = \{g_1, g_2, \ldots\}$ denote the corresponding set of disjoint subgraphs of $\mathcal{G}$ to be transformed to reach $\mathcal{G}'$. Note that for each $g \in G$ there is a $r = (g_l, g_r) \in \Phi_{RR}$ satisfying $g \simeq g_l$. Here, we present a feasible flow of Algorithm II that transforms $\mathcal{G}$ by applying the corresponding r to each $g \in G$. For each $g \in G$, let the nodes in g be active at that time step, and pick a neighbor as illustrated in Fig. 10. Furthermore, let any node that is not included in any $g \in G$ be inactive, which ensures that only the subgraphs in G will be transformed. Finally, let each g pick the corresponding r as the candidate ruler to execute. In that case, the agents are guaranteed to only apply the corresponding $r \in \Phi_{RR}$ to each $g \in G$. Hence, the corresponding transformation has a non-zero probability in P_{RR}, i.e. $P_{RR}(\mathcal{G}\mathcal{G}') > 0$.

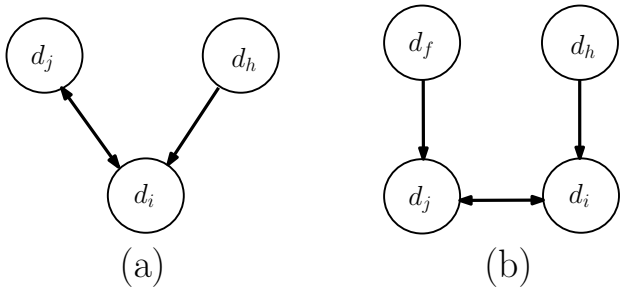


Figure 10. An arrow is pointed from each agent to the neighbor it picked. For each $g \in G$, the nodes in g have non-zero probability to pick their neighbors as shown in (a) if $r = r_1$, and as shown in (b) if $r = r_2$.

■

In general, $\mathbb{G}_{n,m}$ can be represented as the union of two disjoint sets, $\mathbb{G}^0_{n,m}$ (regular graphs) and $\mathbb{G}^+_{n,m}$ (non-regular graphs), defined as

$$\mathbb{G}^0_{n,m} = \{\mathcal{G} \in \mathbb{G}_{n,m} \mid f(\mathcal{G}) = 0\}, \tag{20}$$

$$\mathbb{G}^+_{n,m} = \mathbb{G}_{n,m} \setminus \mathbb{G}^R_{n,m}. \tag{21}$$

Note that if $m \in \mathbb{N}$, then $\mathbb{G}^0_{n,m} \neq \emptyset$ and it is possible to build a regular graph with the available number of edges. In the remainder of this section, we will show that if $\mathbb{G}^0_{n,m} \neq \emptyset$, then P_{RR} has a unique limiting distribution that is uniform over $\mathbb{G}^0_{n,m}$. To this end, first it is shown that all the graphs in $\mathbb{G}^+_{n,m}$ are transient states of P_{RR} if $m \in \mathbb{N}$, i.e. whenever the current graph is non-regular, there is a non-zero probability that the system will leave that configuration and never return back.

Lemma 3.4 *Let $\mathcal{G} \in \mathbb{G}_{n,m}$, and let $m \in \mathbb{N}$. If $\mathcal{G} \in \mathbb{G}^+_{n,m}$, then $\mathcal{G}$ is a transient state of P_{RR} .*

Proof: Let $\mathcal{G} \in \mathbb{G}^+_{n,m}$, and let $m \in \mathbb{N}$. In light of Theorem 2.10, there exists a finite sequence of r_1 applications transforming $\mathcal{G}$ to some $\mathcal{G}' \in \mathbb{G}^0_{n,m}$. Due to Lemma 3.3, the corresponding trajectory is also feasible under P_{RR}. Furthermore, due to Lemma 3.2, it is not possible to return to $\mathcal{G}$ once a regular graph $\mathcal{G}'$ is reached. Hence, $\mathcal{G}$ is transient.

■

Next, it is shown that if $\mathbb{G}^0_{n,m} \neq \emptyset$, then $\mathbb{G}^0_{n,m}$ is a closed communicating class of P_{RR}.

Lemma 3.5 *[42] Let $\mathcal{G} = (V, E)$ be a connected m-regular graph with n nodes and let $m > 2$. Then, in the limit, the Random 1-Flipper operation given below constructs all connected m-regular labeled graphs with the same probability.*

Algorithm III: *Random 1-Flipper:*
1 : *Choose a random edge* $(i, j) \in E$.
2 : *Choose a random node* $h \in \mathcal{N}_i \setminus \{j\}$.
3 : *Choose a random node* $f \in \mathcal{N}_j \setminus \{i\}$.
4 : ***if*** $((i, f) \notin E, (j, h) \notin E)$
5 : $\quad E = (E \setminus \{(i, h), (j, f)\}) \cup \{(i, f), (j, h)\}$.
6 : ***end if***

Proof: This is proved in [42] by showing that a repetitive application of Random 1-Flipper operation (provided below), which is a randomized execution of r_2, induces an aperiodic irreducible Markov chain with a doubly stochastic probability transition matrix over the set of all connected m-regular labeled graphs on n nodes. ■

Lemma 3.6 *If $\mathbb{G}^0_{n,m} \neq \emptyset$, then $\mathbb{G}^0_{n,m}$ is a closed communicating class of P_{RR}.*

Proof: In light of Lemma 3.5, any $\mathcal{G} \in \mathbb{G}^0_{n,m}$ can be reached from any other $\mathcal{G}' \in \mathbb{G}^0_{n,m}$ through a sequence of r_2 applications. Due to Lemma 3.3, the corresponding sequence of r_2 applications is also feasible under P_{RR}. Hence, $\mathbb{G}^0_{n,m}$ is a communicating class. Furthermore, $\mathbb{G}^0_{n,m}$ is closed due to Lemma 3.2. ■

Note that since all the states outside $\mathbb{G}^0_{n,m}$ are transient, $\mathbb{G}^0_{n,m}$ is the only closed communicating class of P_{RR}. Hence, P_{RR} has a unique stationary distribution, μ^*, with the support $\mathbb{G}^0_{n,m}$. In the remainder of this section, the limiting behavior within $\mathbb{G}^0_{n,m}$ is inspected to show that μ^* is a limiting distribution and it is uniform over $\mathbb{G}^0_{n,m}$.

Lemma 3.7 *For any pair of regular graphs, $\mathcal{G}, \mathcal{G}' \in \mathbb{G}^0_{n,m}$,*

$$P_{RR}(\mathcal{G}, \mathcal{G}') = P_{RR}(\mathcal{G}', \mathcal{G}). \tag{22}$$

Proof: For any regular graph, $\mathcal{G} \in \mathbb{G}^0_{n,m}$, r_1 is not applicable on $\mathcal{G}$ since all the nodes have equal degrees. Hence, any transition from $\mathcal{G}$ to another $\mathcal{G}' \in \mathbb{G}^0_{n,m}$ is only via r_2. Note that r_2 is a reversible rule, i.e. if two nodes, i and j, exchange their neighbors, $h \in \mathcal{N}_i$ and $f \in \mathcal{N}_j$, in accordance with r_2, then swapping those neighbors back is also a feasible application of r_2.

Let us consider an arbitrary execution of Algorithm II, where $\mathcal{G} = (V, E) \in \mathbb{G}^0_{n,m}$ is transformed into $\mathcal{G}' = (V, E') \in \mathbb{G}^0_{n,m}$. Let u be the corresponding vector of randomly picked neighbors in line 4 of Algorithm II (let $u_i = \epsilon$ if i was inactive). For each node, i, let $R_i(u)$ be the set of nodes that picked i, and let $M(u) = \{(i, j) \mid i \in R_j(u), j \in R_i(u)\}$ be the set of matched pairs. For $\mathcal{G}'$, consider the vector, u', whose entries are

$$u'_i = \begin{cases} u_i & \text{if } u_i = \epsilon \text{ or } (i, u_i) \in E', \\ u_{u_i} & \text{otherwise.} \end{cases} \tag{23}$$

Note that $\Pr[u] = \Pr[u']$ since the inactive nodes will remain inactive with the same probability, and each active node picks a neighbor uniformly at random. Furthermore,

$M(u) = M(u')$. Also, for every $(i, j) \in M(u)$, we have $|R_i(u)| = |R_i(u')|$ and $|R_j(u)| = |R_j(u')|$. Hence, each $(i, j) \in M(u)$ will reverse the neighbor-swapping in the transition from $\mathcal{G}$ to $\mathcal{G}'$ with the same probability (lines 13-19 in Algorithm II), so $\Pr[\mathcal{G} \to \mathcal{G}'; u] = \Pr[\mathcal{G}' \to \mathcal{G}; u']$. Consequently, $P_{RR}(\mathcal{G}, \mathcal{G}') = P_{RR}(\mathcal{G}', \mathcal{G})$. ■

Theorem 3.8 *Let $\mathbb{G}_{n,m}$ be the state space, and let $m \in \mathbb{N}$. Then, P_{RR} has a unique limiting distribution, μ^*, satisfying*

$$\mu^*(\mathcal{G}) = \begin{cases} 1/|\mathbb{G}^0_{n,m}| & \text{if } \mathcal{G} \in \mathbb{G}^0_{n,m}, \\ 0 & \text{otherwise.} \end{cases} \tag{24}$$

Proof: For $m \in \mathbb{N}$, $\mathbb{G}^0_{n,m} \neq \emptyset$ and $\mathbb{G}^0_{n,m}$ is a closed communicating class due to Lemma 3.6. Furthermore it is the only closed communicating class since all the other states (graphs in $\mathbb{G}^+_{n,m}$) are transient due to Lemma 3.4. As such, P_{RR} has a unique stationary distribution, μ^*, whose support is $\mathbb{G}^0_{n,m}$. In the remainder of the proof, the behavior of the Markov chain within $\mathbb{G}^0_{n,m}$ is inspected to show that μ^* is a limiting distribution and it is uniform over $\mathbb{G}^0_{n,m}$.

Let P^0_{RR} be the $|\mathbb{G}^0_{n,m}|$ by $|\mathbb{G}^0_{n,m}|$ probability transition matrix that only represents the transitions within $\mathbb{G}^0_{n,m}$. Due to Lemma 3.6, P^0_{RR} is irreducible. Also P^0_{RR} is aperiodic since $P^0_{RR}(\mathcal{G}, \mathcal{G}) > 0$ for every $\mathcal{G}$ (for instance, there is a non-zero probability of all the nodes being inactive). As such, P^0_{RR} has a unique limiting distribution, μ^{*0}. Furthermore, due to Lemma 3.7, P^0_{RR} is symmetric, and it is consequently doubly stochastic. As a result, μ^{*0} is uniform over $\mathbb{G}^0_{n,m}$.

■

3.3 Simulation Results

In this section, some simulation results for the proposed scheme are presented. In the first simulation, an arbitrary $\mathcal{G}(0) \in \mathbb{G}^+_{8,3}$, which is illustrated in Fig. 9a, is generated, and the interaction graph is evolved using Algorithm II. In light of Theorem 3.8, the system is expected to approach a uniform limiting distribution over $\mathbb{G}^0_{8,3}$. Note that $\mathbb{G}^0_{8,3}$ is a very large set, hence it is not feasible to track the limiting probability corresponding to each

individual graph in $\mathbb{G}^0_{8,3}$. However, there are only 5 non-isomorphic structures in $\mathbb{G}^0_{8,3}$ as depicted in Fig. 11, and the proportions of time spent in each isomorphism class will be presented instead.

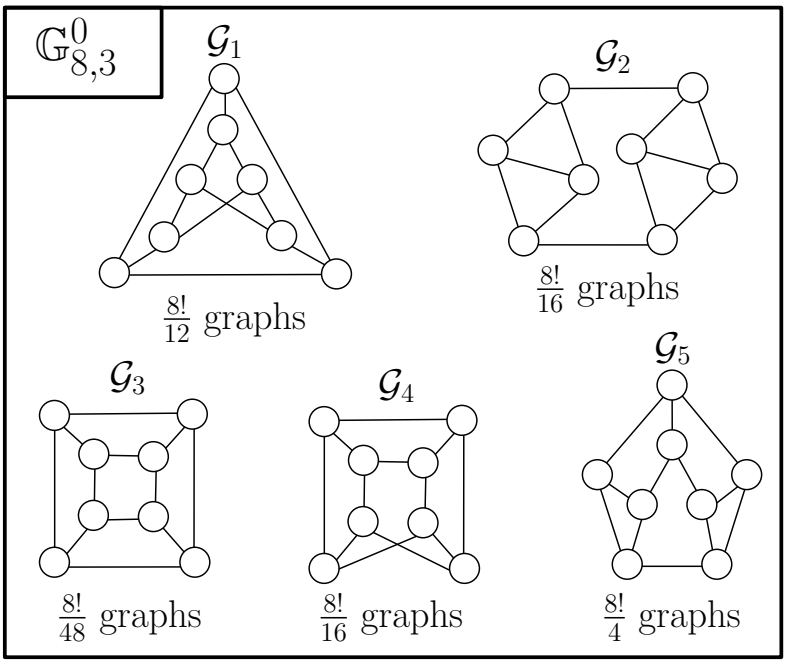


Figure 11. Non-isomorphic graph structures, $\mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_3, \mathcal{G}_4, \mathcal{G}_5$, in $\mathbb{G}^0_{8,3}$, and the number of labeled graphs isomorphic to each structure.

Note that the cardinality of the isomporphism classes in $\mathbb{G}^0_{8,3}$ are not equal since each structure has different symmetry properties. In the simulation, the proportions of the time spent at each of the five isomorphism classes are recorded in a vector $v(t)$ defined as

$$v_i(t) = \frac{|\{0 \leq \tau \leq t \mid \mathcal{G}(\tau) \simeq \mathcal{G}_i\}|}{t+1}, \forall i = 1, \ldots, 5. \tag{25}$$

As the Markov chain approaches the limiting distribution in (49), $v(t)$ is expected to be in alignment with the sizes of isomorphism classes in $\mathbb{G}^0_{8,3}$, i.e. it should approach

$$v^* = [\frac{4}{23}, \frac{3}{23}, \frac{1}{23}, \frac{3}{23}, \frac{12}{23}]. \tag{26}$$

Fig. 12 depicts $\|v(t) - v^*\|_2$ throughout the simulation. Since $\mathcal{G}(0) \in \mathbb{G}^+_{8,3}$, $\|v(t) - v^*\|_2$ is stationary at $\|v^*\|$ for a short period ($t \leq 74$) until the degree range drops to 0, i.e. the

system enters $\mathbb{G}^0_{8,3}$. After that, $\|v(t) - v^*\|_2$ rapidly decreases and approaches 0, as expected from a uniform limiting distribution over $\mathbb{G}^0_{8,3}$.

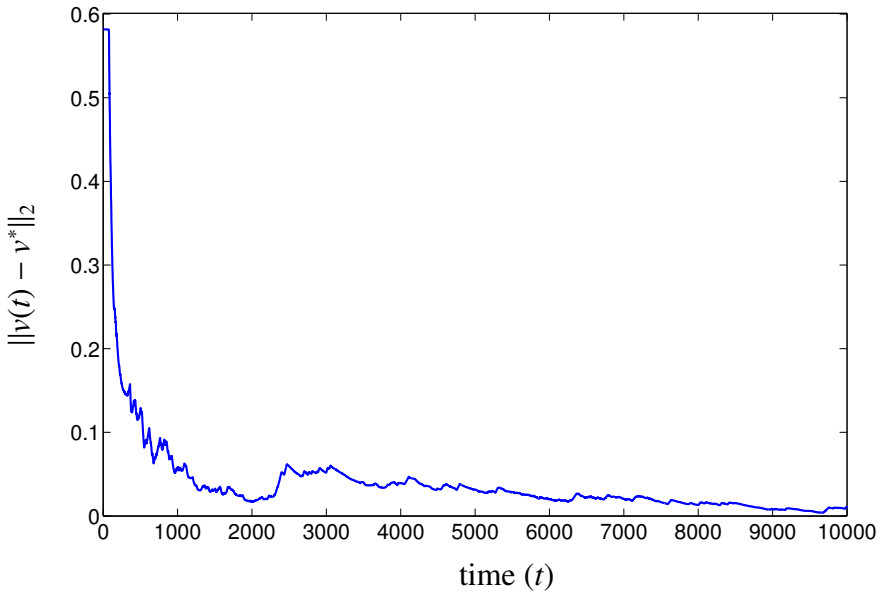


Figure 12. $\|v(t) - v^*\|_2$ **as a function of time.** $v(t)$ **approaches** v^* **in accordance with the uniform limiting distribution over** $\mathbb{G}^0_{8,3}$**.**

In the second simulation, a larger network, $\mathcal{G}(0) \in \mathbb{G}^+_{100,3}$, is considered to illustrate how the robustness of the interaction graph changes under Algorithm II during a period of 10000 time seps. Particularly, the change in the algebraic connectivity is inspected. Fig. 13 illustrates $\mathcal{G}(0)$ and $\mathcal{G}(10000)$, which have algebraic connectivities of 0.032 and 0.195 respectively. On the corresponding trajectory, the degree range drops from 5 to 0 within 2942 steps. After that, the system keeps randomizing within $\mathbb{G}^0_{100,3}$. The evolution of the algebraic connectivity throughout this simulation is shown in Fig. 14. Note that almost every 3-regular graph is almost Ramanujan. Accordingly, the algebraic connectivity of the simulated network is observed to be at least $3 - 2\sqrt{2}$ with a very high probability after a sufficient amount of time.

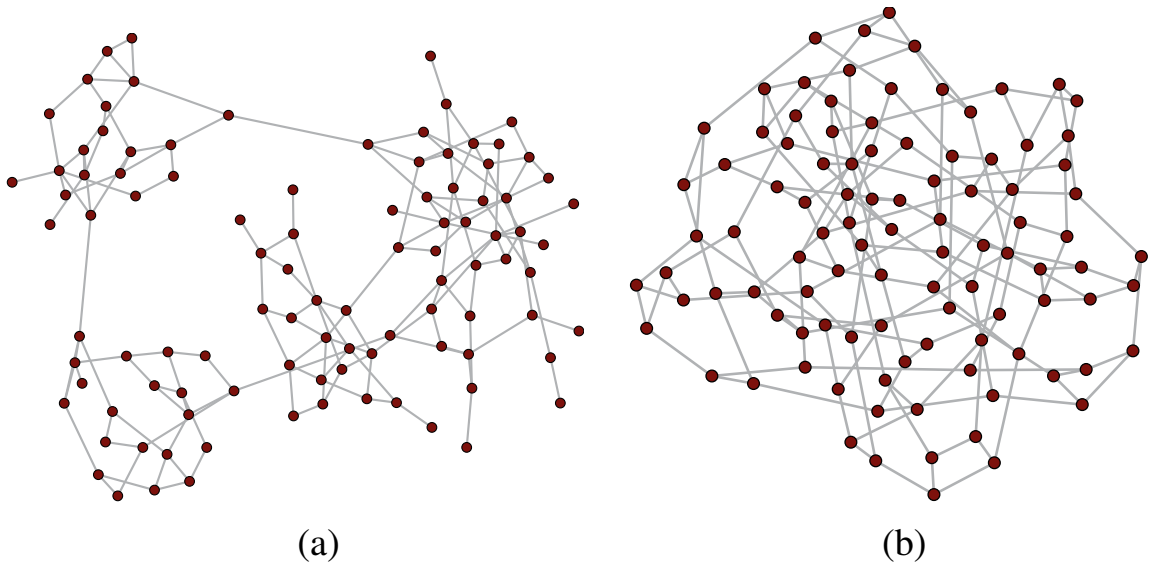


Figure 13. Agents follow Algorithm II so that the initial graph in (a) is transformed into a robust interaction structure such as the one in (b).

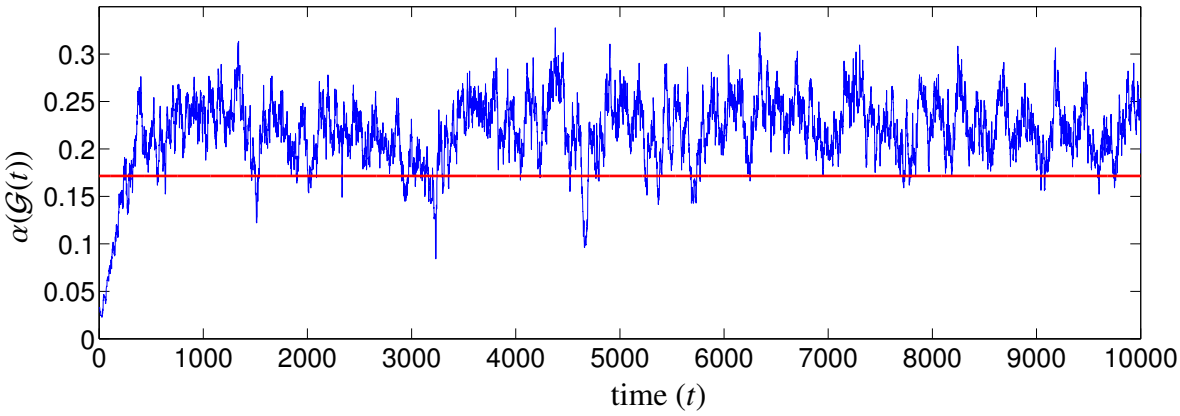


Figure 14. The algebraic connectivity, $\alpha(\mathcal{G}(t))$, as the initial graph in Fig. 13a evolves via Algorithm II. After sufficiently large time, $\alpha(\mathcal{G}(t))$ rarely drops below $3-2\sqrt{2}$ (marked with a solid line), since the corresponding 3-regular graphs are almost Ramanujan with a very high probability.

CHAPTER 4

FORMATION OF RANDOM REGULAR GRAPHS

In this chapter, the randomized degree regularization grammar, Φ_{RR}, is extended to obtain a sparsity-aware decentralized scheme leading to connected random regular graphs, even if the initial average degree is not an integer. In the resulting grammar, Φ^*, the average degree is not necessarily maintained, yet it is guaranteed to remain close to its initial value. In particular, the proposed scheme transforms any connected interaction graph, $\mathcal{G}(0)$, satisfying $\bar{d}(\mathcal{G}(0)) > 2$ into a connected random m-regular graph for some $m \in [\bar{d}(\mathcal{G}(0)), \bar{d}(\mathcal{G}(0)) + 2]$. A distributed implementation, which leads to limiting distributions uniform over the connected m-regular graphs, is also provided.

4.1 Problem Formulation

In the previous chapter, a randomized degree regularization scheme was proposed to obtain random m-regular graphs, which have desirable expansion properties for $m \geq 3$. The corresponding graph grammar, Φ_{RR}, extended the degree regularization grammar, Φ_R, by complementing the degree regularization rule with a neighborhood randomization rule. A distributed implementation was provided in Algorithm II, and it was shown that this algorithm induces a Markov chain, P_{RR}, over the state space, $\mathbb{G}_{n,m}$, consisting of simple connected graphs having the same number of nodes, n, and the same average degree, m, as defined in (18). In Theorem 3.8, it was shown that if the state space, $\mathbb{G}_{n,m}$, consists of graphs with an integer average degree, i.e. $m \in \mathbb{N}$, then P_{RR} has a unique limiting distribution over the m-regular graphs in $\mathbb{G}_{n,m}$. As such, any connected $\mathcal{G}(0)$ satisfying $\bar{d}(\mathcal{G}(0)) \in \mathbb{N}$ is transformed into a connected random $\bar{d}(\mathcal{G}(0))$-regular graph, which is almost surely an expander for $\bar{d}(\mathcal{G}(0)) \geq 3$.

Algorithm II leads to uniformly realized connected random regular graphs for connected initial graphs with integer average degrees. However, its limiting behavior for

graphs with non-integer average degrees requires a further analysis. For such graphs, the degree range converges to 1 due to the regularization rule, as given in Corollary 2.11. Furthermore, the randomization rule keeps randomizing the local neighborhoods. However, an exact characterization of the graphs that will be observed in the limit and their expansion properties are not provided. Note that an initial graph having an integer average degree is a quite restrictive assumption in general since it is a property that can be easily lost due to the removal of even a single node or a single edge. Hence, most systems are unlikely to have an integer average degree unless the interaction graph is designed through a mechanism that strictly enforces that property. Fig. 8 illustrates a pair of graphs on 30 nodes, which are identical except for a single edge. The graph in Fig. 8a has an average degree of 3, whereas the graph in Fig. 8b, which is obtained by removing a single edge from the graph in Fig. 8a, has an average degree of 2.93. Based on the results in the previous chapter, one can say that Algorithm II would transform the graph in Fig. 8a into a random 3-regular graph having desirable expansion properties. However, the results in the previous chapter do not imply high expansion ratios for the graphs that would originate from the graph in Fig. 8b under Algorithm II.

The problem addressed in this chapter is to design a locally applicable sparsity-aware graph transformation scheme that leads to connected random regular graphs, regardless of the initial average degree being an integer or not. Such a solution can transform any connected initial graph into a connected random regular graph with a minimal increase in the overall sparsity. As such, if the initial graph has a sufficient number of edges, the resulting random regular graphs are almost surely expander graphs. Consequently, this chapter presents a generic solution for decentralized formation of random regular graphs in order to transform any connected interaction graph into a robust interaction graph through self-organization.

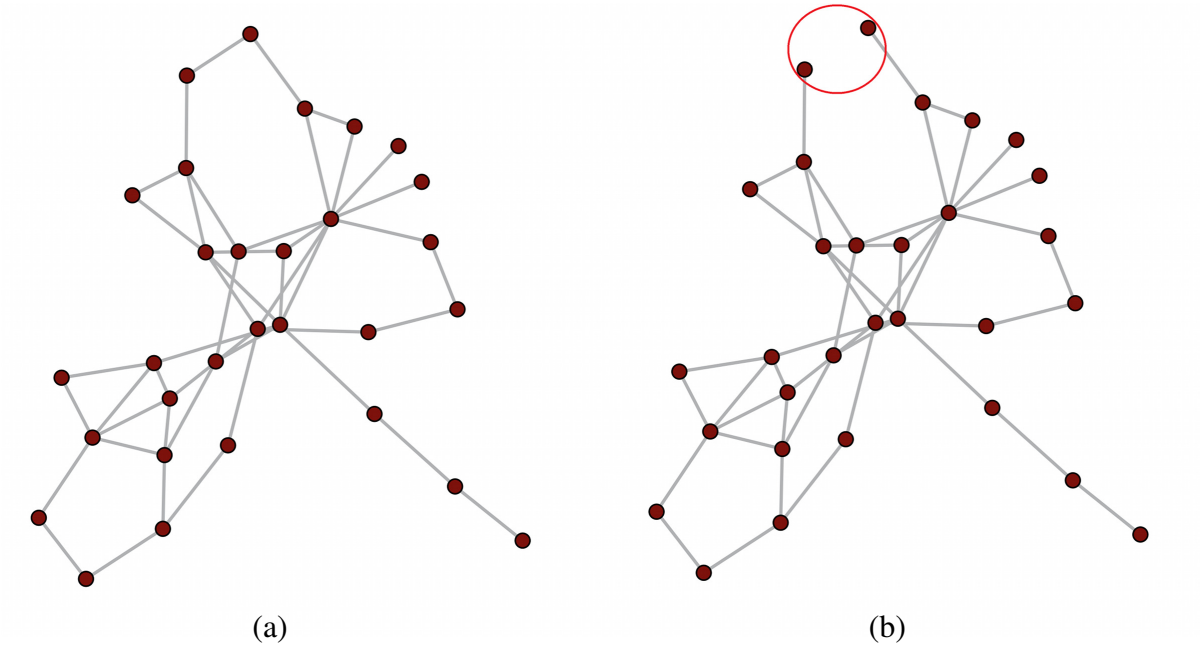


Figure 15. A graph with an integer average degree (a), and a graph with a non-integer average degree (b), which is obtained from the graph in (a) by removing a single edge between the nodes in the circle.

4.2 Proposed Solution

In Chapters 2 and 3, the degree balancing and neighborhood randomization were addressed. In order to incorporate the task of driving the average degree to an integer, the constraint of maintaining the number of edges should be relaxed. In other words, the system should allow for occasional addition and removal of edges as long as the graph is not regular and the total number of edges stays within some proximity of its initial value. Furthermore, the admissible change in the average degree should ensure that there is always a feasible integer value within the reachable interval. In particular, the reachable interval should always contain an even integer since an odd average degree is not feasible for graphs with an odd number of nodes. Furthermore, any possible edge removal should not disconnect the network connectivity. In order to achieve all of these objectives, a graph grammar, Φ^*, is designed by extending Φ_{RR}. The next section presents Φ^* along with a distributed implementation.

4.2.1 Proposed Grammar

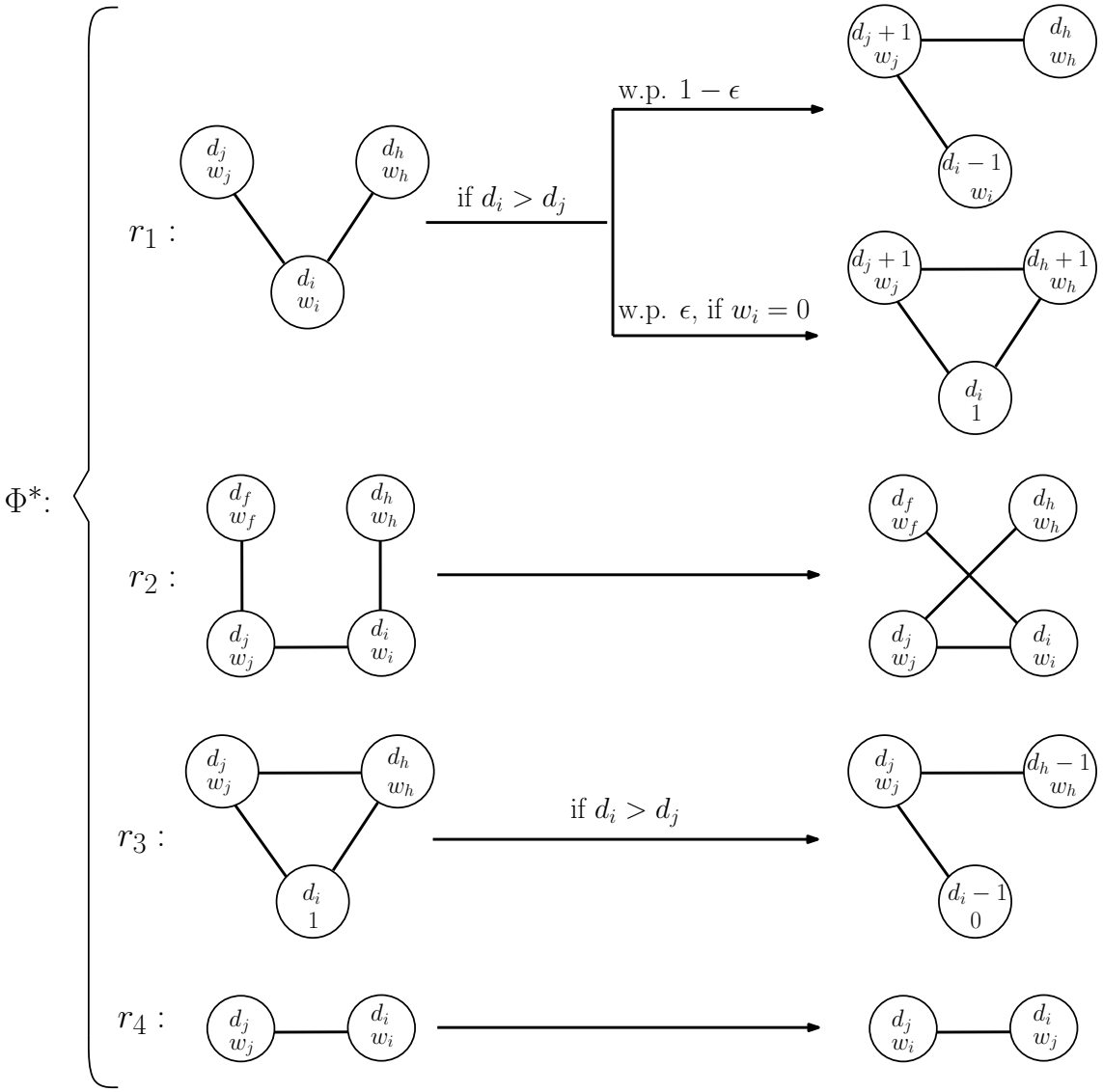
The proposed grammar, $\Phi^* = \{r_1, r_2, r_3, r_4\}$, consists of 4 rules. Different from Φ_R and Φ_{RR}, each node, $i \in V$, has a two dimensional vector, $l(i)$, as its label in Φ^*. The first entry of the label vector is equal to the degree of the node, d_i, just as in Φ_R and Φ_{RR}. The second entry of the label vector is a binary flag, w_i, that denotes whether the corresponding agent is allowed to add or remove an edge. A node, i, can add an edge to the graph only if $w_i = 0$, and it can remove an edge only if $w_i = 1$. As such, the labeling function is a mapping, $l : V \mapsto \{0, 1, 2 \ldots, |V|\} \times \{0, 1\}$, defined as

$$l(i) = \begin{bmatrix} d_i \\ \\ w_i \end{bmatrix}, \forall i \in V. \tag{27}$$

For any graph, $\mathcal{G} = (V, E)$, the degree of each node is already encoded through the edge set, E. Hence, for the sake of simplicity, the first entries of the node labels will be omitted in the notation and a labeled graph will be denoted as

$$\mathcal{G} = (V, E, w), \tag{28}$$

where w is the vector containing the binary entries of the node labels. Using the node labels given in (27), the proposed grammar, $\Phi^* = \{r_1, r_2, r_3, r_4\}$, is defined as



The rules in Φ^* can be interpreted as follows: The first rule, r_1, can be considered as a perturbation of the degree regularization rule in Φ_{RR}. While it behaves identical to the degree regularization rule most of the time, there is also a small probability, ϵ, that if the higher degree node has its second label equal to 0, then it can maintain its link with the neighbor introduced to its lower degree neighbor. Note that the total number of edges increases in that case. The second rule, r_2, is the same neighborhood randomization rule as in Φ_{RR}. The third rule, r_3, is the edge removal rule which breaks one edge of a triangle if all the nodes in that triangle do not have the same degree and one of the higher degree

nodes has its second label equal to 1. Note that the total number of edges decreases in that case. The final rule, r_4, is for exchanging the second labels of adjacent nodes. The purpose of r_4 is to ensure that the edge additions and removals will be applicable as long as they are needed to drive the average degree to an integer. In this setting, the binary labels, w, enables a distributed way of bounding the total number of edges in the network in a close proximity of its initial value throughout the dynamics.

In the remainder of this section, the dynamics induced by Φ^* is analyzed. First, it is shown that Φ^* maintains the graph connectivity.

Lemma 4.1 *Graph connectivity is maintained under* Φ^*.

Proof: All the rules in Φ^* preserve the connectivity of the local structures. Hence, the global connectivity is also maintained under any concurrent application of the rules on disjoint subgraphs. ■

Note that, unlike Φ_R and Φ_{RR}, the the number of edges is not maintained under Φ^* due to the possible edge addition and removals. However, the number of edges is bounded in an interval through the node labels, w.

Lemma 4.2 *Let* $\mathcal{G} = (V, E, w)$ *be a graph, and let* τ *be any feasible trajectory of* $(\mathcal{G}, \Phi^*)$. *For every* $\mathcal{G}' = (V, E', w') \in \tau$,

$$|E'| - |E| = \mathbf{1}^T w' - \mathbf{1}^T w, \tag{29}$$

where $\mathbf{1}$ *is the vector of all ones.*

Proof: Let $\mathcal{G}(t) = (V, E(t), w(t))$ and $\mathcal{G}(t+1) = (V, E(t+1), w(t+1))$ be a pair of consecutive graphs on any trajectory, τ, of $(\mathcal{G}, \Phi^*)$. Note that the number of edges is maintained under r_2 and r_4, and it can increase by 1 only due to r_1 or decrease by 1 only due to r_3. If an application of r_1 adds an extra edge to the network, then one of the nodes involved in that transformation, i, satisfies $w_i(t+1) - w_i(t) = 1$, whereas the other two participating nodes maintain their w entries. On the other hand, for each application of r_3, one of the

nodes involved in that transformation, i, satisfies $w_i(t+1) - w_i(t) = -1$ while the other two participating nodes maintain their w entries. Furthermore, any rule application that maintains the number of edges also maintains the sum of w entries of the participating nodes. In particular, any edge rewiring via r_1 or any neighbor swapping via r_2 maintain the w entries of the participating nodes, whereas any label exchange via r_4 maintains the sum of labels. Hence, $E(t)$ and $E(t+1)$ satisfy

$$|E(t+1)| - |E(t)| = \mathbf{1}^T w(t+1) - \mathbf{1}^T w(t). \tag{30}$$

Using induction, it can be shown that (30) implies (29) for any $\mathcal{G}' = (V, E', w') \in \tau$. ■

Corollary 4.3 *Let $\mathcal{G} = (V, E, \mathbf{0})$ be a graph, and let τ be any feasible trajectory of $(\mathcal{G}, \Phi^*)$. For every $\mathcal{G}' = (V, E', w') \in \tau$,*

$$|E| \leq |E'| \leq |E| + |V|. \tag{31}$$

Proof: Let $\mathcal{G} = (V, E, \mathbf{0})$ be a graph, and let τ be any feasible trajectory of $(\mathcal{G}, \Phi^*)$. In light of Lemma 4.2, every $\mathcal{G}' = (V, E', w') \in \tau$ satisfies

$$|E'| - |E| = \mathbf{1}^T w' - \mathbf{1}^T \mathbf{0} = \mathbf{1}^T w'. \tag{32}$$

Since each entry of w is either equal to 1 or equal 0,

$$0 \leq \mathbf{1}^T w' \leq |V|. \tag{33}$$

Hence, (32) and (33) together imply (31). ■

Corollary 4.3 provides the upper and the lower bounds on the of number of edges along any trajectory of $(\mathcal{G}, \Phi^*)$ for any $\mathcal{G} = (V, E, \mathbf{0})$. Next, it is shown that there is always a number of edges in that interval which implies an integer average degree.

Lemma 4.4 *For any $\mathcal{G} = (V, E)$, a graph, $\mathcal{G}' = (V, E')$, satisfying $\bar{d}(\mathcal{G}') \in \mathbb{N}$ can be formed by adding at most $|V|$ edges to $\mathcal{G}$.*

Proof: Let $\mathcal{G} = (V, E)$ be a graph, and let $\mathcal{G}' = (V, E')$ be a graph that is formed by adding η edges to $\mathcal{G}$. Then,

$$\bar{d}(\mathcal{G}') = \frac{2(|E| + \eta)}{|V|}. \tag{34}$$

For any $\mathcal{G} = (V, E)$, there exists unique pair of integers, $\alpha, \beta \in \mathbb{N}$, such that $0 \leq \beta < |V|$ and

$$|E| = \alpha|V| + \beta. \tag{35}$$

Let $\eta^* = |V| - \beta$. Note that $\eta^* \leq |V|$ since $0 \leq \beta < |V|$. Furthermore, by plugging η^* into (34), we obtain

$$\frac{2(|E| + \eta^*)}{|V|} = \frac{2(|E| + |V| - \beta)}{|V|} = 2\alpha + 2\frac{\beta}{|V|} + 2 - 2\frac{\beta}{|V|} = 2(\alpha + 1). \tag{36}$$

Since $\alpha \in \mathbb{N}$, we have $2(\alpha + 1) \in \mathbb{N}$. Hence, for any $\mathcal{G} = (V, E)$, there exists $0 \leq \eta \leq \eta^*$ such that a graph, $\mathcal{G}' = (V, E')$, satisfying $\bar{d}(\mathcal{G}') \in \mathbb{N}$ can be obtained by adding η edges to $\mathcal{G}$. ■

In light of Lemma 4.4, the interval in (31) always has at least one value implying an integer average degree. Next, it is shown that $(\mathcal{G}, \Phi^*)$ almost surely reaches a regular graph for any connected $\mathcal{G} = (V, E, \mathbf{0})$ such that $\bar{d}(\mathcal{G}) > 2$.

Lemma 4.5 *Let $\mathcal{G} = (V, E, w)$ be a connected graph such that $\bar{d}(\mathcal{G}) \notin \mathbb{N}$. If $\mathbf{1}^T w < |V|$, then $\mathcal{G}$ can be transformed via Φ^* into a graph, $\mathcal{G}' = (V, E', w')$, satisfying $|E'| = |E| + 1$.*

Proof: If $\bar{d}(\mathcal{G}) \notin \mathbb{N}$, then $\mathcal{G}$ has to be a non-regular graph. Furthermore, since $\mathcal{G}$ is connected, then there has to be at least a pair of nodes, i and j, with different degrees. Without loss of generality let $d_i > d_j$. Then, i should have at least one neighbor that is not linked to j. As such, r_1 is applicable on $\mathcal{G} = (V, E)$. Furthermore, since each entry of w is either 0 or 1, at least one entry of w is equal to 0 if $\mathbf{1}^T w < |V|$. If $w_j = 0$, then an extra edge can be added to the graph via r_1 with probability ϵ. If $w_i \neq 0$, then there exists another node, i', such that $w_{i'} = 0$. Apply r_4 along the shortest path between i and i' to obtain $w_i = 0$. Once $w_i = 0$ is obtained, then with probability ϵ an extra edge can be added to the graph in accordance with r_1 to obtain a graph, $\mathcal{G}' = (V, E', w')$, satisfying $|E'| = |E| + 1$. ■

Lemma 4.6 *Let $\mathcal{G} = (V, E, w)$ be a connected triangle-free graph. If $\bar{d}(\mathcal{G}) > 2$, then $\mathcal{G}$ can be transformed via Φ^* into a graph, $\mathcal{G}' = (V, E', w)$, containing at least one triangle.*

Proof: Let $\mathcal{G} = (V, E, w)$ be a connected triangle-free graph. Since $\bar{d}(\mathcal{G}) > 2$, $\mathcal{G}$ has to be a cyclic graph. Consider the shortest cycle on $\mathcal{G}$, C_s. Note that C_s has to contain at least 4 nodes, since $\mathcal{G}$ is triangle-free. Furthermore, if $\bar{d}(\mathcal{G}) > 2$, then C_s cannot contain all the nodes of $\mathcal{G}$, as otherwise $\mathcal{G}$ has to be a cycle graph with $\bar{d}(\mathcal{G}) = 2$. Also, since $\mathcal{G}$ is a connected graph, at least one node on C_s must be linked to a node that is not contained in C_s. Without loss of generality, let i be a node on C_s connected to an off-cycle node, h. Furthermore, let j denote a neighbor of i on C_s. Note that h can not be connected to j since $\mathcal{G}$ is triangle-free. Furthermore, since the C_s contains at least 4 nodes, j has to have another neighbor on C_s, f, which is not connected to i. If i and j execute r_2 by rewiring h and f to each other, C_s becomes a shorter cycle. Note that w is invariant under r_2. Hence, by repeating this process until C_s consists of 3 nodes, $\mathcal{G}$ can be transformed into a graph, $\mathcal{G}' = (V, E', w)$, containing at least one triangle. ■

Lemma 4.7 *Let $\mathcal{G} = (V, E, w)$ be a connected graph such that $\bar{d}(\mathcal{G}) \notin \mathbb{N}$. If $\mathbf{1}^T w > 0$ and $\bar{d}(\mathcal{G}) > 3$, then $\mathcal{G}$ can be transformed via Φ^* into a graph, $\mathcal{G}' = (V, E', w')$, satisfying $|E'| = |E| - 1$.*

Proof: Let $\mathcal{G} = (V, E, w)$ be a connected graph such that $\bar{d}(\mathcal{G}) \notin \mathbb{N}$. Since $\bar{d}(\mathcal{G}) \notin \mathbb{N}$, the degree range can be reduced to 1 without changing $\bar{d}$ and w via r_1, as given in Corollary 2.11. Furthermore, in light of Lemma 4.6, if the resulting graph is triangle-free, then it can be transformed via r_2 into a graph, $\mathcal{G}^+ = (V, E^+, w)$, which contains at least one triangle. Since $\Delta(\mathcal{G}^+) - \delta(\mathcal{G}^+) = 1$ and $\bar{d}(\mathcal{G}^+) = \bar{d}(\mathcal{G}) > 3$, we have $\delta(\mathcal{G}^+) \geq 3$.

If $\mathbf{1}^T w > 0$, then at least one entry of w is equal to 1. As such, $w_i = 1$ can be reached for any $i \in V$ via a sequence of r_4 applications. Hence, if any of the triangles in $\mathcal{G}^+$ involves nodes with non-uniform degrees, then one edge of the triangle can be broken in accordance with r_3 to obtain a graph, $\mathcal{G}' = (V, E')$, satisfying $|E'| = |E| - 1$.

On the other hand, assume that all the triangles on $\mathcal{G}^+$ consist of nodes with uniform degrees. Let i, j, and h be the nodes of such a triangle. Consider the shortest path from any node of this triangle to a node whose degree is not equal to the degrees of the triangle nodes. Without loss of generality, let this path be between i and i' such that $d^+_{i'} \neq d^+_i$. Note that each node other than i' on this path has its degree equal to d^+_i, and either $d^+_i > d^+_{i'}$ or $d^+_i < d^+_{i'}$. Both cases are inspected below:

1) If $d^+_i > d^+_{i'}$, then $d^+_i \geq 4$ given that $\delta(\mathcal{G}^+) \geq 3$. Hence, the node next to i' on the shortest path has at least one neighbor outside the shortest path that is not connected to i'. Applying r_1 by rewiring such a neighbor to i' results in a shorter path from i to a smaller degree node. Applying the same procedure eventually results in a smaller degree node being adjacent to i. Since $d^+_i \geq 4$, i has at least one neighbor other than j and h to rewire to its smaller degree neighbor. After the corresponding application of r_1, we obtain a triangle of nodes with non-uniform degrees. Hence, one edge of the triangle can be broken in accordance with r_3 to obtain a graph, $\mathcal{G}' = (V, E', w')$, satisfying $|E'| = |E| - 1$.

2) If $d^+_i < d^+_{i'}$, then i' can apply r_1 to rewire one of its neighbors to the node on the shortest path next to itself. This results in a shorter path from i to higher degree node is obtained. Applying the same procedure eventually increases the degree of i, and we obtain a triangle of nonuniform degrees. Hence, one edge of the triangle can be broken in accordance with r_3 to obtain a graph, $\mathcal{G}' = (V, E', w')$, satisfying $|E'| = |E| - 1$. ■

Lemma 4.8 *Let $\mathcal{G} = (V, E, \mathbf{0})$ be a connected graph. If $\bar{d}(\mathcal{G}) > 2$, then $(\mathcal{G}, \Phi^*)$ almost surely reaches an m-regular graph such that $\bar{d}(\mathcal{G}) \leq m \leq \bar{d}(\mathcal{G}) + 2$.*

Proof: Let $\mathcal{G} = (V, E, \mathbf{0})$ be a connected graph such that $\bar{d}(\mathcal{G}) > 2$. Due to Corollary 4.3, the number of edges stays in $\{|E|, |E| + 1, \ldots |E| + |V|\}$ along any trajectory of $(\mathcal{G}, \Phi^*)$. In light of Lemma 4.4, this interval contains at least one value, $|E| + \eta$, such that the corresponding average degree, m, is an integer satisfying

$$\bar{d}(\mathcal{G}) \leq m = \frac{2(|E| + \eta)}{|V|} \leq \bar{d}(\mathcal{G}) + 2. \tag{37}$$

As such, $m \geq 3$ since $\bar{d}(\mathcal{G}) > 2$. Let $\mathcal{G}' = (V, E', w')$ be a graph reached from $\mathcal{G}$ via Φ^* such that $\bar{d}(\mathcal{G}') \notin \mathbb{N}$. Then, either $|E'| < |E| + \eta$ or $|E'| > |E| + \eta$. Both cases are inspected below:

1) If $|E'| < |E| + \eta$, then $\mathbf{1}^T w' = |E'| - |E| < |V|$. As such, in light of Lemma 4.5, $\mathcal{G}'$ can have its number of edges increased by 1 via Φ^*. This process can be repeated unless the average degree reaches an integer value.

2) If $|E'| > |E| + \eta$, then $\mathbf{1}^T w' = |E'| - |E| > 0$. Furthermore, $\bar{d}(\mathcal{G}') > 3$ since $m \geq 3$. As such, in light of Lemma 4.7, $\mathcal{G}'$ can have its number of edges decreased by 1 via Φ^*. This process can be repeated unless the average degree reaches an integer value.

Hence, in either case it is possible to add or remove edges to the network via Φ^* until the number of edges implies an integer average degree, $m \in [\bar{d}(\mathcal{G}), \bar{d}(\mathcal{G}) + 2]$. Once an average degree of m is obtained, then the graph can be driven to a m-regular configuration through a sequence of r_1 applications. As such, any $\mathcal{G}'$ has a non-zero probability of reaching an m-regular graph after a finite sequence of Φ^* applications. Consequently, $(\mathcal{G}, \Phi^*)$ almost surely reaches an m-regular graph such that $\bar{d}(\mathcal{G}) \leq m \leq \bar{d}(\mathcal{G}) + 2$. ■

Note that, unlike Φ_R and Φ_{RR}, the degree range does not monotonically decrease under Φ^* due to the possible edge addition and removals. However, if a regular graph is reached, then the graph remains regular.

Lemma 4.9 *Let $\mathcal{G}$ be an m-regular graph. Any feasible trajectory of $(\mathcal{G}, \Phi^*)$ consists of only m-regular graphs.*

Proof: Since $\mathcal{G}$ is a m-regular graph, $d_i = d_j$ for any pair of nodes, i, j. Hence, r_1 and r_3 are not applicable on any regular graph. Furthermore, since the node degrees are invariant to the applications of r_2 and r_4, any feasible trajectory of $(\mathcal{G}, \Phi^*)$ consists of only m-regular graphs. ■

Once an m-regular graph is reached, the graph evolves only via the randomization rules,

r_2 and r_4. In the next section, a distributed implementation of Φ^*, which leads to uniform limiting distributions over the connected m-regular graphs, is presented.

4.2.2 Distributed Implementation

In general, there may be many feasible applications of Φ^* on an interaction graph. In such cases, the agents need to randomly execute Φ^* on disjoint subgraphs without any global coordination. To this end, Algorithm IV is proposed as a distributed implementation of Φ^*. By following Algorithm IV, the nodes can concurrently modify their local neighborhoods in accordance with Φ^* such that any feasible transformation occurs with a non-zero probability.

Note that Algorithm IV is memoryless since each iteration only depends on the current graph, and the probability of any feasible transition is independent of the past. Furthermore, connectivity is maintained due to Lemma 4.1. Furthermore, for any connected initial graph, $\mathcal{G}(0) = (V, E(0), \mathbf{0})$, the average degree remains in $[\bar{d}(\mathcal{G}), \bar{d}(\mathcal{G}) + 2]$ due to Corollary 4.3. Let k denote the average degree of the initial graph, i.e.

$$k = \bar{d}(\mathcal{G}(0)). \tag{38}$$

Algorithm IV induces a Markov chain over the state space, $\mathbb{G}_{n,[k,k+2]}$, consisting of simple connected labeled graphs with n nodes and average degrees contained in $[k, k+2]$, i.e.

$$\mathbb{G}_{n,[k,k+2]} = \{\mathcal{G} = (V, E, w) \mid |V| = n, |E| = 0.5k|V| + \mathbf{1}^T w, w_i \in \{0, 1\} \forall i \in V\}. \tag{39}$$

Let $\mu(t)$ denote the probability distribution over the state space at time t. Then, $\mu(t)$ satisfies

$$\mu^T(t+1) = \mu^T(t) P^*, \tag{40}$$

where P^* is the corresponding probability transition matrix. Accordingly, the probability of transition from any $\mathcal{G}$ to any $\mathcal{G}'$ is denoted by $P^*(\mathcal{G}, \mathcal{G}')$. Note that for any pair of graphs, $\mathcal{G}, \mathcal{G}' \in \mathbb{G}_{n,[k,k+2]}$, the corresponding transition probability under Algorithm IV, $P^*(\mathcal{G}, \mathcal{G}')$, is non-zero if and only if it is possible to transform $\mathcal{G}$ into $\mathcal{G}'$ in a single time step via Φ^*.

Algorithm IV: Distributed Implementation of Φ^*

initialize: $\mathcal{G} = (V, E, w = \mathbf{0})$ connected, $\epsilon \in (0, 1)$ small
repeat
  Each agent, i, is active with probability $1 - \epsilon$.
  Each active i picks a random $j \in \mathcal{N}_i$.
  For each $i \in V$, $R_i = \{i' \in \mathcal{N}_i \mid i' \text{ picked } i\}$.
  for (each (i, j) s.t. $i \in R_j$, $j \in R_i$, $d_i \geq d_j$)
   $\max\{i, j\}$ picks a random $r \in \Phi^*$.
    if ($r = r_1$, $d_i > d_j$, $|R_i| \geq 2$)
      i picks a random $h \in R_i \setminus \{j\}$.
      if $((j, h) \notin E)$
        i picks a random $\epsilon' \in [0, 1]$.
        if ($\epsilon' \geq \epsilon$)
          $E = (E \setminus \{(i, h)\}) \cup \{(j, h)\}$.
        else if ($w_i = 0$)
          $E = E \cup \{(j, h)\}$, $w_i = 1$.
        end if
      end if
    else if ($r = r_2$, $|R_i| \geq 2$,$|R_j| \geq 2$)
      i picks a random $h \in R_i \setminus \{j\}$.
      j picks a random $f \in R_j \setminus \{i\}$.
      if $((i, f) \notin E, (j, h) \notin E)$
        $E = (E \setminus \{(i, h), (j, f)\}) \cup \{(i, f), (j, h)\}$.
    else if ($r = r_3$, $|R_i| \geq 2$, $w_i = 1$)
      i picks a random $h \in R_i \setminus \{j\}$.
      if $((j, h) \in E)$
        $E = E \setminus \{(i, h)\}$, $w_i = 0$.
      end if
    else if ($r = r_4$)
      Swap w_i and w_j.
    end if
  end for
end repeat

Lemma 4.10 *Let $\mathcal{G}, \mathcal{G}' \in \mathbb{G}_{n,[k,k+2]}$ be any pair of graphs. $P^*(\mathcal{G}, \mathcal{G}') > 0$ if and only if $\mathcal{G}'$ can be reached from $\mathcal{G}$ in one step via Φ^*.*

Proof: $\Rightarrow$: Since any possible transformation in Algorithm IV (lines 13, 15, 22, 26, 29) satisfies a rule in Φ^*, $P^*(\mathcal{G}, \mathcal{G}') > 0$ implies that $\mathcal{G}'$ can be reached from $\mathcal{G}$ in one step via Φ^*.

$\Leftarrow$: Let $\mathcal{G}'$ be reachable from $\mathcal{G}$ in one step via Φ^*, and let $G = \{g_1, g_2, \ldots\}$ denote the corresponding set of disjoint subgraphs of $\mathcal{G}$ to be transformed to reacg $\mathcal{G}'$. Note that for each $g \in G$ there is a $r = (g_l, g_r) \in \Phi^*$ satisfying $g \simeq g_l$. Here, a feasible flow of Algorithm IV that transforms $\mathcal{G}$ by applying the corresponding r to each $g \in G$ is presented. For each $g \in G$, let the nodes in g be active at that time step, and pick a neighbor as illustrated in Fig. 16. Furthermore, let any node that is not included in any $g \in G$ be inactive, which ensures that only the subgraphs in G will be transformed. Finally, let each g pick the corresponding r as the candidate ruler to execute. In that case, the agents are guaranteed to only apply the corresponding $r \in \Phi^*$ to each $g \in G$. Hence, the corresponding transformation has a non-zero probability in P^*.

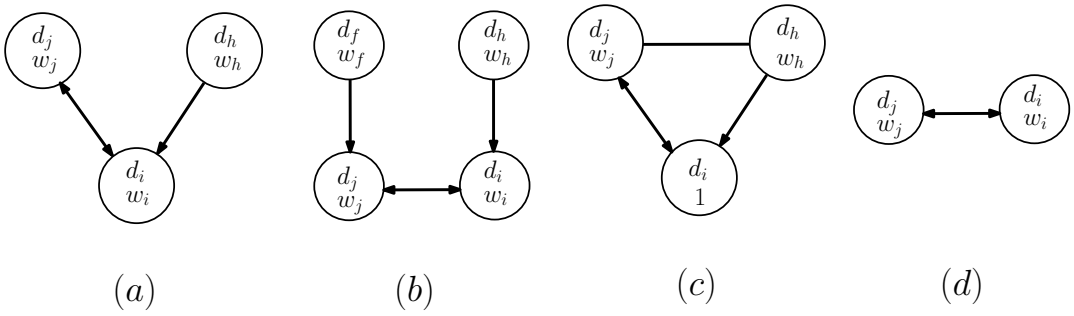


Figure 16. An arrow is pointed from each agent to the neighbor it picked. For each $g \in G$, the nodes in g have non-zero probability to pick their neighbors as shown in (a) if $r = r_1$, (b) if $r = r_2$, (c) if $r = r_3$, and (d) if $r = r_4$.

■

The state space, $\mathbb{G}_{n,[k,k+2]}$ can be represented as the union of two disjoint sets, $\mathbb{G}^0_{n,[k,k+2]}$ (regular graphs) and $\mathbb{G}^+_{n,[k,k+2]}$ (non-regular graphs), defined as

$$\mathbb{G}^0_{n,[k,k+2]} = \{\mathcal{G} \in \mathbb{G}_{n,[k,k+2]} \mid f(\mathcal{G}) = 0\}, \tag{41}$$

$$\mathbb{G}^+_{n,[k,k+2]} = \mathbb{G}_{n,[k,k+2]} \setminus \mathbb{G}^0_{n,[k,k+2]}. \tag{42}$$

In light of Lemma 4.8, $\mathbb{G}^0_{n,[k,k+2]} \neq \emptyset$ for $k > 2$. Let M denote the set of all such m, i.e.

$$M = \{m \in \mathbb{N} \mid k \leq m \leq k+2, 0.5m|V| \in \mathbb{N}\}. \tag{43}$$

Based on the values in M, the set of regular graphs, $\mathbb{G}^0_{n,[k,k+2]}$, can be written as the union of some disjoint sets as

$$\mathbb{G}^0_{n,[k,k+2]} = \bigcup_{m \in M} \mathbb{G}^0_{n,m}, \tag{44}$$

where each $\mathbb{G}^0_{n,m}$ is the set of m-regular graphs defined as

$$\mathbb{G}^0_{n,m} = \{\mathcal{G} \in \mathbb{G}^0_{n,[k,k+2]} \mid \bar{d}(\mathcal{G}) = m\}. \tag{45}$$

In the remainder of this section, the limiting behavior of P^* is analyzed. Note that in light of Lemma 4.8, if $k > 2$, then any connected initial graph, $\mathcal{G}(0) = (V, E(0), \mathbf{0})$, almost surely converges to $\mathbb{G}^0_{n,[k,k+2]}$. Next, it is shown that, for each $m \in M$, $\mathbb{G}^0_{n,m}$ is a closed communicating class.

Lemma 4.11 *If $k > 2$, then $\mathbb{G}^0_{n,m}$ is a closed communicating class of P^* for each $m \in M$.*

Proof: In Light of Lemma 4.9, only r_2 and r_4 is applicable on any m-regular graph, and any resulting graph is m-regular. Hence, once the system reaches any $\mathbb{G}^0_{n,m}$, it stays in $\mathbb{G}^0_{n,m}$. Furthermore, in light of Lemma 3.5, any m-regular structure can be reached from any other via r_2. Moreover, since all the graphs in $\mathbb{G}^0_{n,m}$ are connected, any permutation of the elements in w is also reachable via r_1 for each graph structure in $\mathbb{G}^0_{n,m}$. Hence, each $\mathbb{G}^0_{n,m}$ is a closed communicating class of P^*. ■

The limiting behavior of Algorithm IV depends on the limiting behavior of P^* in each closed communicating class. Note that for each $\mathbb{G}^0_{n,m}$, there is a unique stationary distribution, μ^*_m, of P^* whose support is $\mathbb{G}^0_{n,m}$. Next, it is shown that each μ^*_m is a limiting distribution that is uniform over $\mathbb{G}^0_{n,m}$. To this end, first it is shown that the transitions between any two regular graphs are symmetric.

Lemma 4.12 *For any $m \in M$, let $\mathcal{G}, \mathcal{G}' \in \mathbb{G}^0_{n,m}$ be any pair of graphs. Then,*

$$P^*(\mathcal{G}, \mathcal{G}') = P^*(\mathcal{G}', \mathcal{G}). \tag{46}$$

Proof: For any $m \in M$, let $\mathcal{G}, \mathcal{G}' \in \mathbb{G}^0_{n,m}$ be any pair of graphs. Note that r_1 and r_3 are not applicable on any graph in $\mathbb{G}^0_{n,m}$ since all the nodes have equal degrees. Hence, any transition from $\mathcal{G}$ to $\mathcal{G}'$ is only via r_2 or r_4. Note that both r_2 and r_4 are reversible. Hence, $P^*(\mathcal{G}, \mathcal{G}') > 0$ if and only if $P^*(\mathcal{G}', \mathcal{G}) > 0$.

Let us consider an arbitrary execution of Algorithm IV, where $\mathcal{G}$ is transformed into $\mathcal{G}'$. Let u be the corresponding vector of randomly picked neighbors in line 4 of Algorithm IV (let $u_i = \epsilon$ if i is inactive). For each node, i, let $R_i(u)$ be the set of nodes that picked i, and let $M(u) = \{(i, j) \mid i \in R_j(u), j \in R_i(u)\}$ be the set of matched pairs. In the remainder of the proof, it is shown that for any such feasible execution there exists an equally likely execution of Algorithm IV that transforms $\mathcal{G}'$ back to $\mathcal{G}$. For $\mathcal{G}' = (V, E', w')$, consider the vector, u', whose entries are

$$u'_i = \begin{cases} u_i & \text{if } u_i = \epsilon \text{ or } (i, u_i) \in E', \\ u_{u_i} & \text{otherwise.} \end{cases} \tag{47}$$

Note that $\Pr[u] = \Pr[u']$ since the inactive nodes will remain inactive with the same probability, and each active node picks a neighbor uniformly at random. Furthermore, $M(u) = M(u')$. Let each $(i, j) \in M(u)$ randomly choose the same candidate rule, r, they executed in the transition from $\mathcal{G}$ to $\mathcal{G}'$. As such, if $r = r_4$, then i and j will swap w_i and w_j back. On the other hand, if $r = r_2$, then i and j will reverse the neighbor-swapping in

the transition from $\mathcal{G}$ to $\mathcal{G}'$ with the same probability (lines 12-17 in Algorithm IV) since $|R_i(u)| = |R_i(u')|$ and $|R_j(u)| = |R_j(u')|$ for every $(i, j) \in M(u)$. As such, all the local transformations from $\mathcal{G}$ to $\mathcal{G}'$ can be reversed with the same probability under Algorithm IV. Consequently, $P^*(\mathcal{G}, \mathcal{G}') = P^*(\mathcal{G}', \mathcal{G})$. ■

Lemma 4.13 *For any $m \in M$, there is a unique limiting distribution, μ^*_m, of P^* satisfying*

$$\mu^*_m(\mathcal{G}) = \begin{cases} 1/|\mathbb{G}^0_{n,m}| & \text{if } \mathcal{G} \in \mathbb{G}^0_{n,m}, \\ 0 & \text{otherwise.} \end{cases} \tag{48}$$

Proof: For any $m \in M$, $\mathbb{G}^0_{n,m}$ is a closed communicating class due to Lemma 4.11. As such, for each $\mathbb{G}^0_{n,m}$, there exists a unique stationary distribution, μ^*_m, whose support is $\mathbb{G}^0_{n,m}$. Let P^*_m be the $|\mathbb{G}^0_{n,m}|$ by $|\mathbb{G}^0_{n,m}|$ probability transition matrix that only represents the transitions within $\mathbb{G}^0_{n,m}$. Due to Lemma 4.11, P^*_m is irreducible. Also P^*_m is aperiodic since $P^*(\mathcal{G}, \mathcal{G}) > 0$ for every $\mathcal{G}$ (for instance, there is a non-zero probability of all the nodes being inactive). As such, the corresponding stationary distribution, μ^*_m, is a limiting distribution. Furthermore, due to Lemma 4.12, P^*_m is symmetric, and it is consequently doubly stochastic. As a result, μ^*_m is uniform over $\mathbb{G}^0_{n,m}$.

■

Theorem 4.14 *For any connected initial graph, $\mathcal{G} = (V, E, \mathbf{0})$, satisfying $\bar{d}(\mathcal{G}) > 2$, P^* leads to a limiting distribution, $\mu^*_{\mathcal{G}}$, given as*

$$\mu^*_{\mathcal{G}} = \sum_{m \in M} \Pr[\mathcal{G} \to \mathbb{G}^0_{n,m}] \mu^*_m, \tag{49}$$

where $\Pr[\mathcal{G} \to \mathbb{G}^0_{n,m}]$ is the probability that the chain initialized at $\mathcal{G}$ ever enters the set of m-regular graphs, $\mathbb{G}^0_{n,m}$, and each μ^*_m is uniform over its support, $\mathbb{G}^0_{n,m}$.

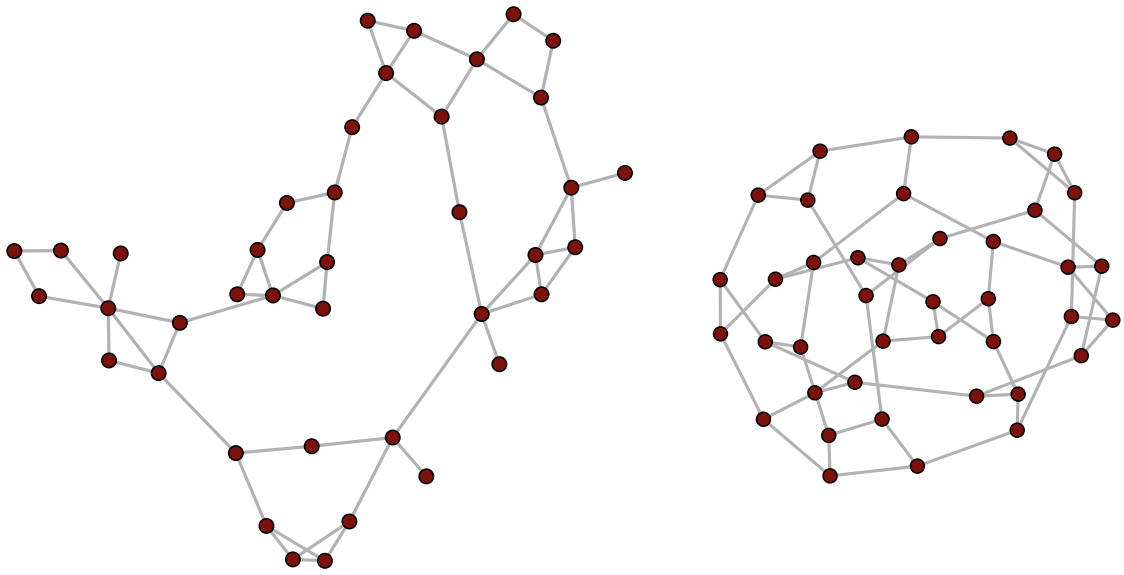
Proof: In light of Lemma 4.8, any connected $\mathcal{G} = (V, E, \mathbf{0})$, satisfying $\bar{d}(\mathcal{G}) > 2$ almost surely enters a closed communicating class of m-regular graphs, $\mathbb{G}^0_{n,m}$, such that $m \in M$. Furthermore, each $\mathbb{G}^0_{n,m}$ has a uniform limiting distribution, μ^*_m as given in Lemma 4.13. Hence, P^* leads to the convex combination of limiting distributions, μ^*_m, where each

μ^*_m is weighted by $\Pr[\mathcal{G} \to \mathbb{G}^0_{n,m}]$, which is the probability that the chain starting at $\mathcal{G}$ ever enters $\mathbb{G}^0_{n,m}$. ■

In light of Theorem 4.14, Algorithm IV asymptotically transforms any connected initial graph, $\mathcal{G} = (V, E, \mathbf{0})$, satisfying $\bar{d}(\mathcal{G}) > 2$ into a random m-regular graph for some $m \in M$ as given in (43). When M is not a singleton, then the probability of reaching each $\mathbb{G}^0_{n,m}$ depends on the initial graph. Note that $\bar{d}(\mathcal{G}) > 2$ implies $m \geq 3$ for every $m \in M$. Hence, for any connected initial graph, $\mathcal{G} = (V, E, \mathbf{0})$, satisfying $\bar{d}(\mathcal{G}) > 2$, the graphs observed in the limit are almost Ramanujan with a very high probability.

4.3 Simulation Results

In this section, we present some simulation results for the proposed scheme. In the first simulation, an arbitrary connected graph, $\mathcal{G}(0) = (V, E, \mathbf{0})$, with 40 nodes and 56 edges. As such, the initial average degree is $\bar{d}(\mathcal{G}(0)) = 2.8$. The interaction graph is evolved using Algorithm IV with $\epsilon = 0.05$ for a period of 10000 steps. The initial and the final graphs for this simulation are illustrated in Fig. 17. Since $\bar{d}(\mathcal{G}(0)) = 2.8$, the average degree is guaranteed to remain in $[2.8, 4.8]$ for any feasible trajectory, as given in Corollary 4.3. Since the number of nodes is even, both integers in this interval are feasible values for the average degree, i.e. $M = \{3, 4\}$. In light of Theorem 4.14, the graph is expected to become either a random 3-regular graph or a random 4-regular. In the simulation presented here, the graph reaches a 3-regular graph after 862 time steps. Note that once the system enters $\mathbb{G}^0_{40,3}$, both $f(\mathcal{G}(t))$ and $\bar{d}(\mathcal{G}(t))$ are stationary. The values of the degree range, $f(\mathcal{G}(t))$, and the average degree, $\bar{d}(\mathcal{G}(t))$, for the first 1000 steps are illustrated in Fig. 18. Once a 3-regular graph is reached, the graph keeps randomizing in $\mathbb{G}^0_{40,3}$ via r_2 and r_4. The evolution of the algebraic connectivity throughout this simulation is shown in Fig. 19. Note that almost every 3-regular graph is almost Ramanujan. Accordingly, the algebraic connectivity of the simulated network is observed to be at least $3 - 2\sqrt{2}$ with a very high probability after a sufficient amount of time.



(a) (b)

Figure 17. Agents follow Algorithm IV so that the initial graph in (a) is transformed into a robust interaction structure such as the one in (b).

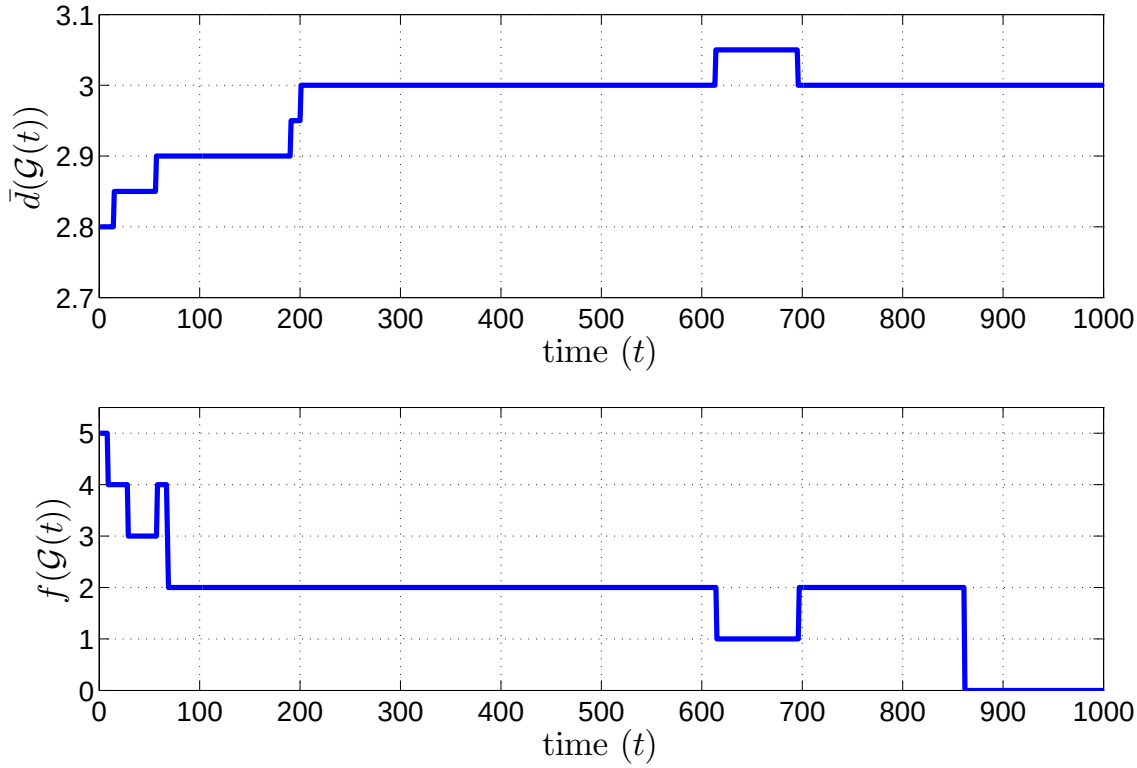


Figure 18. The average degree, $\bar{d}(\mathcal{G}(t))$, and the degree range, $f(\mathcal{G}(t))$, for the first 1000 time steps. Once $f(\mathcal{G}(t)) = 0$ is reached, both $\bar{d}(\mathcal{G}(t))$ and $f(\mathcal{G}(t))$ remains stationary under Φ^*.

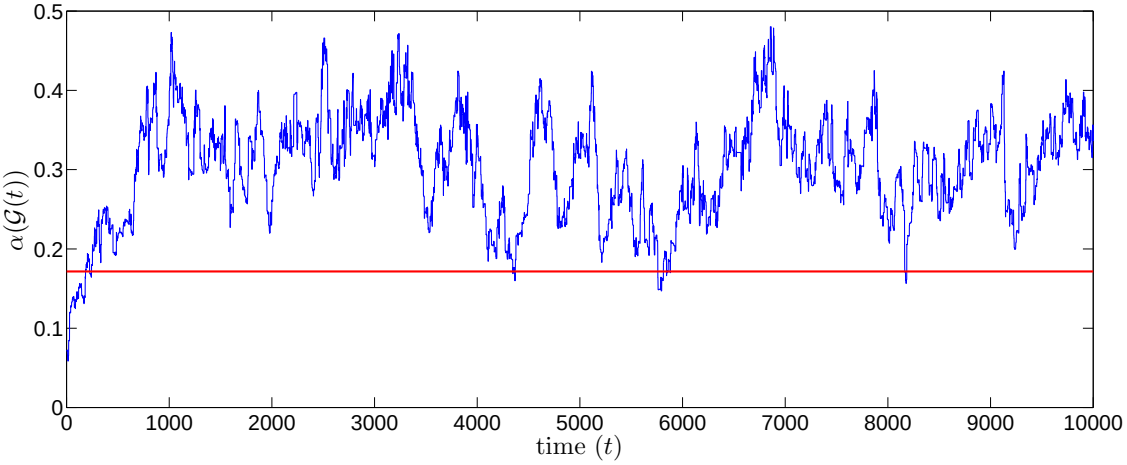


Figure 19. The algebraic connectivity, $\alpha(\mathcal{G}(t))$, as the initial graph in Fig. 17a evolves via Algorithm IV. After sufficiently large time, $\alpha(\mathcal{G}(t))$ rarely drops below $3 - 2\sqrt{2}$ (marked with a solid line), since the corresponding 3-regular graphs are Ramanujan with a very high probability.

CHAPTER 5

DISTRIBUTED GRAPH COVERAGE

This chapter presents a decentralized scheme for optimal protection of networks by a group of mobile security resources with local capabilities. In particular, the following research problem is considered: Assume that some resources with local monitoring and protection capabilities are arbitrarily deployed on a network that is unkown apriori. How can these resources explore the network and optimize their locations in a decentralized manner to efficiently protect the overall network? This problem is formulated as a distributed coverage control problem on a graph, and a decentralized solution is obtained using a game theoretic framework. The network is modeled as a connected graph, and the resources are modeled as a group of mobile agents with possibly heterogeneous sensing and communication capabilities. The agents are initially deployed on some arbitrary nodes of the graph, and their task is to maximize the number of covered nodes. The agents are assumed to have distance limitations in their sensing and communication capabilities. Each agent covers the nodes within its sensing range, and the set of those nodes are only known to the other agents within its communication range. Furthermore, in the corresponding discrete-time system, the agents are allowed to move locally, i.e. they either maintain their positions or move to some adjacent nodes at each time step.

In the remainder of this chapter, the distributed graph coverage problem is formulated, and a solution to this problem is obtained using a game theoretic framework. Specifically, a corresponding potential game is designed and a learning algorithm is employed to maximize the overall coverage. Using the proposed scheme, the agents can asymptotically maintain maximum coverage with an arbitrarily high probability.

5.1 Problem Formulation

Consider a connected undirected graph, $\mathcal{G} = (V, E)$, and let $I = \{1, 2, \ldots, m\}$ denote a set of m mobile agents located on some nodes, $\{v_1, v_2, \ldots, v_m\} \subseteq V$. Let each agent, $i \in I$, have a sensing range, δ_i. We assume that each agent, i, can sense the subgraph induced by the nodes within its δ_i-neighborhood. As such, an agent $i \in I$ located at a node $v_i \in V$ covers all the nodes in $\mathcal{N}_{v_i}^{\delta_i}$ given as

$$\mathcal{N}_{v_i}^{\delta_i} = \{v \in V \mid d(v, v_i) \leq \delta_i\}, \tag{50}$$

where $d(v, v_i)$ denotes the distance between v and v_i. Any node of the graph is covered if it is included in the δ-neighborhood of at least one agent, and the set of covered nodes, $V_c \subseteq V$, is given as

$$V_c(v_1, \ldots, v_m) = \bigcup_{i \in I} \mathcal{N}_{v_i}^{\delta_i}. \tag{51}$$

Furthermore, let each agent, i, have a communication range, δ_i^c. At each time step, each agent, i, broadcasts the list of nodes covered by itself to the other agents within its communication range, δ_i^c. Through such communications, the set of nodes covered by i is known to j, if j is at most δ_i^c away from i, as illustrated in Fig. 20. Agents update their positions on the graph based on the available information. Each agent can either maintain its position or move to an adjacent node in the consecutive time steps of the corresponding discrete time system.

Given a set of mobile agents with local capabilities, the goal in the distributed graph coverage (DGC) problem is to design some local rules for the agents to follow in updating their positions such that they asymptotically maximize the number of covered nodes, $|V_c(t)|$. In general, a rule is considered to be local if its execution by an agent requires only some information available within a small distance from the agent. Following such local rules, the agents can move on the graph in a decentralized fashion.

Definition *(Distributed Graph Coverage Problem)*: Let m mobile agents be deployed on a connected graph, $\mathcal{G} = (V, E)$. For each agent i, let $v_i(t) \in V$ denote its position, and let $V_c(t)$ denote the set of covered nodes. Find some local rules for updating $v_1(t), \ldots, v_m(t)$ to asymptotically maximize $|V_c(t)|$ subject to $d(v_i(t+1), v_i(t)) \leq 1$ for every i.

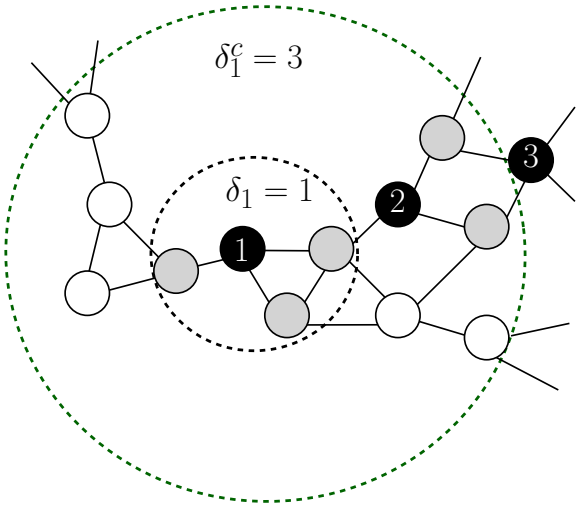


Figure 20. An illustration of the agent capabilities in the DGC problem. Agent 1 has as cover range $\delta_1 = 1$ and communication range $\delta_1^c = 3$. The set of nodes that are covered by agent 1 is known to agent 2 via local communications. However, agent 3 does not receive that information.

5.2 Solution Approach

In the DGC problem, a group of mobile agents are required to explore a graph, which is unknown apriori, and to cover as many nodes of the graph as possible. As such, the underlying locational optimization problem is similar to the maximum coverage problem in [65] with uniformly weighted nodes. Such NP-hard problems are typically tackled by finding sufficiently good approximate solutions through fast algorithms (e.g. [81, 82, 69]). Similarly, in many distributed coverage control studies, a locational optimization function is optimized by the agents aiming for the best local improvements (e.g., [51]-[56]). Such a distributed greedy approach can be employed to solve the DGC problem. Accordingly, the agents may move on the graph to maximally improve the local coverage. However, the resulting performance would significantly depend on the graph structure and the initial

configuration. A distributed greedy method may quickly lead to a good approximation if the agents start with a sufficiently good initial coverage or if the interaction graph has certain structural properties. However, it may lead to very inefficient configurations for arbitrary graphs and initial conditions. For instance, consider the scenario in Fig. 21, where 2 agents with sensing ranges of 1 can achieve globally optimal coverage in 2 time steps. In this example, the agents would not leave the initial configuration through a greedy approach, and the resulting performance would be arbitrarily poor for an arbitrarily larger graph obtained by adding more nodes attached to the initially uncovered hub.

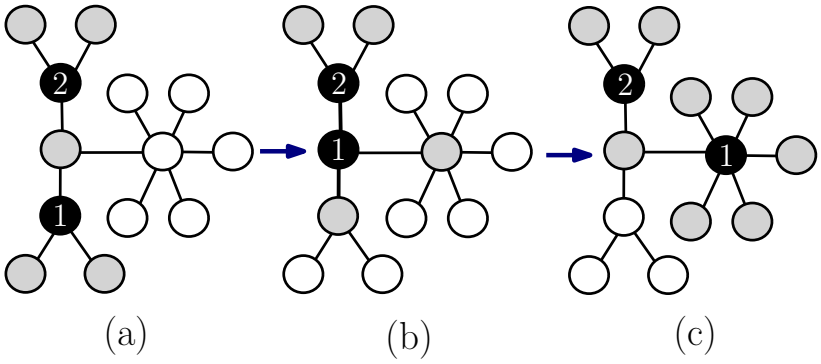


Figure 21. A possible trajectory to a globally optimal configuration in a simple example. Two agents with cover ranges of 1 are initially located as in (*a*). The number of covered nodes (shown in gray and black) is reduced in the intermediate step in (b) to reach the global optimum in (c).

In order to ensure efficient coverage for arbitrary graphs and initial configurations, a solution method should occasionally allow for graph exploration at the expense of a better coverage. In this work, we present such a solution by approaching the problem from a game theoretic perspective. In the resulting scheme, while the agents have high probabilities of taking actions that locally improve coverage, they may also take locally worse actions with much smaller probabilities in order to allow for further exploration and search for a global optimum. As such, the proposed method is essentially similar to algorithms such as ϵ-greedy, softmax, and simulated annealing (e.g., [83, 75]), whereas the distributed nature of the system is captured via a game theoretic formulation. Also, through the utility design and the constrained action sets, both the information required by the agents and the position updates are local in the resulting scheme.

Before presenting the proposed game theoretic solution, some preliminaries about game theory and stochastic stability are provided.

5.2.1 Game Theory Preliminaries

A finite strategic game $\Gamma = (I, A, U)$ consists of three components: (1) a set of m players (agents) $I = \{1, 2, \ldots, m\}$, (2) an m-dimensional action space $A = A_1 \times A_2 \times \ldots \times A_m$, where each A_i is the action set of player i, and (3) a set of utility functions $U = \{U_1, U_2, \ldots, U_m\}$, where each $U_i : A \mapsto \Re$ is a mapping from the action space to real numbers. For any action profile, $a \in A$, let a_{-i} denote the actions of players other than i. Using this notation, an action profile a can also be represented as $a = (a_i, a_{-i})$. Each utility function represents the preferences of the corresponding agent over the action profiles. In strategic games, each agent aims to optimize its own utility. A well known notion of equilibrium for such utility maximizers is Nash equilibrium. An action profile, $a^* \in A$, is called a Nash equilibrium if each agent is playing a best response to the actions of the other agents, i.e.

$$U_i(a_i^*, a_{-i}^*) = \max_{a_i} U_i(a_i, a_{-i}^*), \quad \forall i \in I \tag{52}$$

Game theoretic formulation of a cooperative control problem involves the design of a corresponding game and a learning algorithm. In game theoretic learning, starting with an arbitrary action profile, the agents repetitively play a game. At each time instant $t \in \{0, 1, 2, \ldots\}$, each agent $i \in I$ plays an action $a_i(t)$ and receives some utility, $U_i(a(t))$. Based on the history of received utilities, the agents update their actions by following a learning algorithm. Some learning algorithms include, but not limited to, Cournot best response, fictitious play, and log-linear learning. In this setting, the role of learning is to drive the action profile to the set of desired action profiles. The dynamics of the action profile induced by a learning algorithm depends on the utility functions. For instance, if all agents follow the Cournot best response, then the action profile almost surely converges to a pure Nash equilibrium if the utility functions imply the finite improvement property, i.e. for any action profile there is a finite sequence of unilateral best response updates leading to a Nash

equilibrium.

A class of games that is widely utilized in cooperative control problems is potential games. In potential games, there exists a potential function $\phi : A \mapsto \mathfrak{R}$ such that the change of a player's utility resulting form its unilateral deviation from an action profile equals the resulting change in ϕ. More precisely, for each player i, for every $a_i, a_i' \in A_i$, and for all $a_{-i} \in A_{-i}$,

$$U_i(a_i', a_{-i}) - U_i(a_i, a_{-i}) = \phi(a_i', a_{-i}) - \phi(a_i, a_{-i}). \tag{53}$$

In cooperative control applications, usually the potential function ϕ represents the global objective of the control problem, which depends on the actions of all agents. Note that (53) and (52) together imply that the set of potential maximizers is a subset of the Nash equilibria. Once a cooperative control problem is mapped to a potential game, various game theoretic learning algorithms, such as log-linear learning (LLL) [84], can be utilized to drive the action profile to the set of potential maximizers. Essentially, the LLL is a noisy best response algorithm. It induces a Markov chain over the action space with a limiting distribution, μ_ϵ^*, where ϵ denotes the noise parameter. As the noise parameter, ϵ, goes down to zero, the limiting distribution, μ_ϵ^*, has an arbitrarily large part of it mass over the set of potential maximizers. This result pertains to the notion of stochastic stability, which is presented next.

5.2.2 Stochastic Stability Preliminaries

Definition *(Regular Perturbed Markov Chain)*: Let P be the transition matrix of a discrete-time Markov chain over a finite state space $\mathcal{X}$. A perturbed Markov chain with the noise parameter ϵ is called a regular perturbed Markov chain if

1. P_ϵ is aperiodic and irreducible for $\epsilon > 0$,

2. $\lim_{\epsilon \to 0} P_\epsilon = P$,

3. For any $x, x^+ \in \mathcal{X}$ if $P_\epsilon(x, x^+) > 0$, then

$$0 < \lim_{\epsilon \to 0^+} \frac{P_\epsilon(x, x^+)}{\epsilon^{R(x,x^+)}} < \infty, \tag{54}$$

where $R(x, x^+)$ is called the resistance of the corresponding transition.

Since a regular perturbed Markov chain, P_ϵ, is aperiodic and irreducible, it has a limiting distribution μ_ϵ^*.

Definition *(Stochastically Stable State)*: Let P_ϵ denote a regular perturbed Markov chain over a state space, $\mathcal{X}$. Any state, $x \in \mathcal{X}$, is stochastically stable if

$$\lim_{\epsilon \to 0^+} \mu_\epsilon^*(x) > 0. \tag{55}$$

5.3 Proposed Solution

In this section, the proposed game theoretic solution for the DGC problem is presented.

5.3.1 Game Design

In order to formulate the DGC problem in a game theoretic setting, first a corresponding game should be designed. In particular, a potential game, $\Gamma_{DGC} = (I, A, U)$, will be designed in this section by defining the action sets and the utility functions.

In the DGC problem, the coverage provided by each agent is determined by the position of the agent. Hence, the action of an agent can be defined as its position on the graph. As such, each action set is equal to the node set of $\mathcal{G} = (V, E)$, i.e.

$$A_i = V, \quad \forall i \in I. \tag{56}$$

Accordingly, an action profile, $a \in A$, is the vector of agent positions.

In order to obtain a potential game that can be used to solve the DGC problem, the number of covered nodes is chosen to be the corresponding potential function $\phi(a)$, i.e.

$$\phi(a) = |V_c(a)|. \tag{57}$$

Then, the utilities should be designed such that $\phi(a)$ in (57) is indeed a potential function for the resulting game. This can be achieved by setting the agent utility functions as

$$U_i(a) = |\mathcal{N}_{a_i}^{\delta_i} \setminus \bigcup_{j \neq i} \mathcal{N}_{a_j}^{\delta_j}|. \tag{58}$$

As such, for any action profile, each agent gathers a utility equal to the number of nodes that are only covered by itself. Note that this utility is equal to the marginal contribution of the corresponding agent to the number of covered nodes. An action profile and the corresponding utilities are illustrated through an example in Fig. 22, where two agents with communication ranges of 1 are covering a graph consisting of 6 nodes.

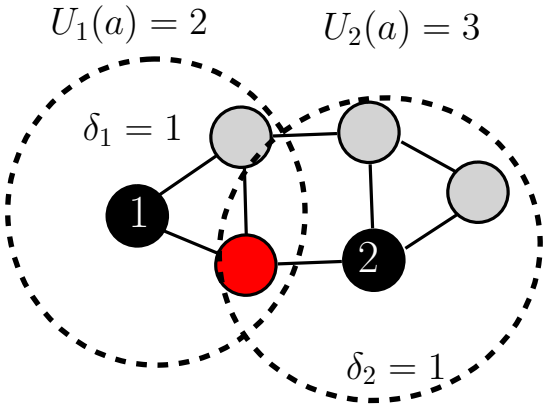


Figure 22. Two agents are covering a graph. In Γ_{DGC}, the action of an agent is its position on the graph, and each agent gathers a utility equal to the number of nodes that are only covered by itself. Hence, none of the agents gathers a utility for covering the node shown in red.

Lemma 5.1 *Utility functions in (58) lead to a potential game $\Gamma_{DGC} = (P, A, U)$ with the potential function given in (57).*

Proof: Let $a_i = v_i$ and $a_i' = v_i'$ be two possible actions for agent i, and let a_{-i} denote the actions of other agents. Due to (51) and (57),

$$\phi(a) = |\bigcup_{i \in I} \mathcal{N}_{a_i}^{\delta_i}| \tag{59}$$

For any agent i, (59) can be expanded as

$$\phi(a) = |\mathcal{N}_{a_i}^{\delta_i} \setminus \bigcup_{j \neq i} \mathcal{N}_{a_j}^{\delta_j}| + |\bigcup_{j \neq i} \mathcal{N}_{a_i}^{\delta_i}| = U_i(a_i, a_{-i}) + |\bigcup_{j \neq i} \mathcal{N}_{a_j}^{\delta_j}|. \tag{60}$$

Using (60) for any pair of actions a_i and a_i',

$$\phi(a_i', a_{-i}) - \phi(a_i, a_{-i}) = U_i(a_i', a_{-i}) - U_i(a_i, a_{-i}). \tag{61}$$

■

5.3.2 Learning Dynamics

In order to solve the DGC problem through Γ_{DGC}, a learning algorithm should be employed to drive the agent to the action profiles that maximize the number of covered nodes. Note that Γ_{DGC} is designed such that the number of covered nodes is the potential function.

For potential games, a learning algorithm known as log-linear learning (LLL) can be used to asymptotically reach action profiles that maximize the potential function $\phi(a)$ [84]. LLL induces a regular perturbed Markov chain over the action space. Furthermore, the stochastically stable states are the global maximizers of the potential function, i.e.

$$\lim_{\epsilon \downarrow 0} \mu_\epsilon^*(a) > 0 \iff \phi(a) \geq \phi(a'), \forall a' \in A, \tag{62}$$

where μ_ϵ^* is the limiting distribution of the resulting Markov chain, and $\epsilon > 0$ is the noise parameter to be set in the LLL algorithm.

However, the classical LLL assumes that each player i has access to all the actions in its action set A_i. In general, the stochastic stability of the potential maximizers is not guaranteed when the system evolves on constrained action sets, i.e. when each agent i is allowed to choose its next action action $a_i(t+1)$ only from a subset of actions $A_i^c(a_i(t))$ that depends on its current action $a_i(t)$. Note that this is indeed the case for the DGC problem, where each agent can either maintain its position or move to an adjacent node, i.e.

$$A_i^c(a_i) = \{a_i' \mid d(a_i, a_i') \leq 1\}, \tag{63}$$

where $d(a_i, a_i')$ denotes the distance between a_i and a_i'.

The issue of constrained action sets was addressed in [85], and Binary Log-Linear Learning (BLLL) was proposed as a variant of LLL that can maximize the potential in a setting with constrained action updates.

Algorithm V: Binary Log-linear Learning ([85])
1 : **initialization:** $\epsilon \in \Re^+$ small, $a \in A$ arbitrary
2 : **repeat**
3 : Pick a random $i \in I$, and a random $a_i' \in A_i^c(a_i)$.
4 : Compute $\alpha = \epsilon^{-U_i(a(t))}, \beta = \epsilon^{-U_i(a_i', a_{-i}(t))}$.
5 : With probability $\frac{\beta}{\alpha+\beta}$, set $a_i = a_i'$.
6 : **end repeat**

In BLLL, a single agent is randomly chosen at each time step. The selection of a single agent at each time step can be achieved (with a very high probability) without a centralized coordination. One possible distributed approach is to use the asynchronous time model proposed in [86]. In this model, each agent has a virtual clock that ticks according to a rate 1 Poisson process, and an agent is chosen whenever its clock ticks. As such, with a very high probability, a single updating agent is picked at each step. The selected agent, assuming that all the other agents are stationary, updates its action depending on its current utility and the hypothetical utility it would receive by playing a random action in its constrained action set. This is illustrated in Fig. 23.

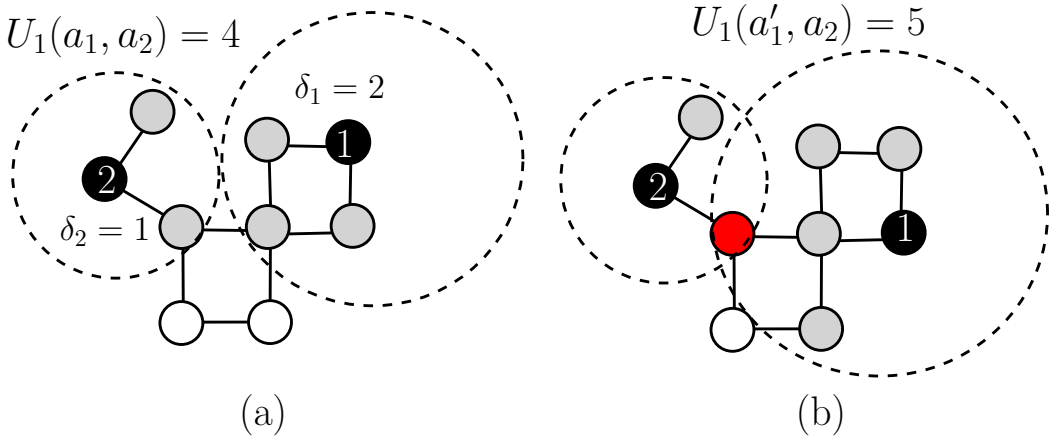


Figure 23. An illustration of the BLLL algorithm. Two agents have the action profile, (a_1, a_2), as in (a), and agent 1 is updating its action. Agent 1 randomly picks a candidate action, $a_1' \in A_i^c(a_i)$, as in (b). Its next action is picked from $\{a_1, a_1'\}$ with probabilities depending on the corresponding utilities.

In [85], it was shown that BLLL can be used to achieve potential maximization in a constrained setting if the constrained action sets satisfy Properties 1 and 2 provided below.

Property 1 (Reachability) For any agent $p_i \in P$ and any action pair $a_i^0, a_i^k \in A_i$, there exists a sequence of actions $\{a_i^0, a_i^1, \ldots, a_i^k\}$ such that $a_i^r \in A_i^c(a_i^{r-1})$ for all $r \in \{1, 2, \ldots, k\}$.

Property 2 (Reversability) For any agent $p_i \in P$ and any action pair $a_i, a_i' \in A_i$,

$$a_i' \in A_i^c(a_i) \iff a_i \in A_i^c(a_i').$$

Theorem 5.2 *[85] Consider any finite potential game and constrained action sets satisfying Properties 1 and 2. If all players adhere to binary log-linear learning, then the stochastically stable states are the set of potential maximizers.*

In light of Theorem 5.2, the agents can maximize the coverage by following the BLLL algorithm in a repetitive play of Γ_{DGC}, if the constrained action sets given in (63) satisfy Properties 1 and 2. Lemma 5.3 shows that the constrained action sets in Γ_{DGC} indeed satisfy these properties if the graph to be covered is connected.

Lemma 5.3 *The constrained action sets in (63) satisfy Properties 1 and 2 if the graph $\mathcal{G} = (V, E)$ is connected.*

Proof: If the graph is connected, then there exists a finite-length path $\{v^0, \ldots, v^k\}$ between any pair of nodes $v^0, v^k \in V$, and Property 1 is satisfied. Furthermore, for undirected graphs, $d(v, v') = d(v', v)$. Hence, Property 2 is also satisfied. ■

5.3.3 Sufficient Communications

In order to execute BLLL in a repetitive play of Γ_{DGC}, each agent should be able to compute its current utility as well as the hypothetical utilities it may gather by moving to a neighboring node. As such, the agents should be able to compute those utilities by using the information gathered through local communications. This requirement is satisfied if the agents have sufficiently large communication ranges. Accordingly, a sufficient condition for the communication ranges is derived as follows.

Lemma 5.4 *An agent, $i \in I$, is within the communication range of any other agent covering a node, $v \in V$, if*

$$d(a_i, v) \leq \min_{j \in I}(\delta_j^c - \delta_j), \tag{64}$$

where δ_j and δ_j^c denote the cover range and the communication range of j, respectively.

Proof: Let k be an agent covering v. From the triangle inequality,

$$d(a_i, a_k) \leq d(a_i, v) + d(a_k, v). \tag{65}$$

Note that $d(a_k, v) \leq \delta_k$ since v is covered by k. Using this inequality and plugging (64) into (65),

$$d(a_i, a_k) \leq \min_{j \in I}(\delta_j^c - \delta_j) + \delta_k. \tag{66}$$

Since $\min_{j \in I}(\delta_j^c - \delta_j) \leq \delta_k^c - \delta_k$,

$$d(a_i, a_k) \leq \delta_k^c - \delta_k + \delta_k = \delta_k^c, \tag{67}$$

which implies that i is within the communication range of k. ■

Lemma 5.5 *Each agent, i, can compute $U_i(a_i', a_{-i})$ for any action $a_i' \in A_i^c(a_i)$ if*

$$\delta_j^c - \delta_j \geq \delta^* + 1, \quad \forall j \in I, \tag{68}$$

where $\delta^* = \max_{j \in I} \delta_j$.

Proof: If (68) holds, then

$$\min_{j \in I}(\delta_j^c - \delta_j) \geq \delta^* + 1. \tag{69}$$

Lemma 5.4 and (69) together imply that i located at a_i knows the number of agents covering any node v satisfying

$$d(a_i, v) \leq \delta^* + 1. \tag{70}$$

In light of (63), an updating agent, i, can at most be 1 hop away from its current position, a_i, in the next time step. Hence, its coverage set in the next time step can only contain nodes

that are at most $\delta_i + 1$ away from its current position. Note that any such node, v, satisfies (70) since $\delta_i \leq \delta^*$ for any agent p_i. Hence, i can compute $U_i(a_i', a_{-i})$ for any $a_i' \in A_i^c(a_i)$. ■

Theorem 5.6 *Let each agent, $i \in I$, has a cover range δ_i and a communication range δ_i^c satisfying (68). If all agents follow BLLL in a repetitive play of Γ_{DGC} subjected to the constraint in (63), then the stochastically stable action profiles are the coverage maximizers.*

Proof: If (68) is satisfied, then Lemma 5.5 implies that the agents can compute the hypothetical utilities for all the actions in (63). Hence, the agents can follow BLLL to update their actions. In that case, from Theorem 5.2 and Lemma 5.3, only the potential maximizers are stochastically stable in a repetitive play of Γ_{DGC}. Due to (57), those are the configurations maximizing the number of covered nodes, $|V_c(a)|$. ■

5.4 Simulation Results

In this section, some simulation results for the proposed method are presented. In the simulations, a group of agents are initially deployed on an arbitrary node of a connected graph. The graph consists of 50 nodes and 78 edges, and it has a diameter of 17. Two simulations, one for a group of homogeneous agents and one for a group of heterogeneous agents, are presented below. In both simulations, the agents start with the same initial condition, and they follow the BLLL algorithm with $\epsilon = 0.015$. For both simulations, the number of covered nodes for a period of 10000 time steps, and the configuration of the agents on the graph at some instants are presented.

In the first simulation, there are 13 homogeneous agents. Each agent has a cover range of 1 and a communication range of 3. Initially, only 5 nodes are covered by the agents. The number of covered nodes throughout the simulation is shown in Fig. 24. In accordance with the BLLL algorithm, the agents are more likely to choose actions that improve their local coverage, whereas locally worse actions can also be taken with much smaller probabilities to avoid any possible convergence to a poor local optima. Accordingly, the number of covered nodes is non-decreasing most of the time, but it can occasionally decrease as

well. In the simulation, the agents rapidly spread out on the graph, and they cover more than ninety percent of the graph within 2000 time steps. Once a reasonable coverage is achieved, reaching the global optima, i.e. a complete coverage, takes relatively longer. After a sufficient amount of time, the agents maintain a complete coverage most of the time due to the stochastic stability of the potential maximizers. The configuration of the agents on the graph at some instants of this simulation are provided in Fig. 25.

In the second simulation, there are 10 heterogeneous agents. Seven agents have cover ranges of 1 and communication ranges of 4, whereas the remaining three agents have cover ranges of 2 and communication ranges of 5. Agents start at the same node as the previous simulation, and initially they cover 10 nodes. The number of covered nodes throughout the simulation is shown in Fig. 26. The agents rapidly spread out on the graph, and they cover more than ninety percent of the graph within 2500 time steps. After 4200 time steps, the agents maintain a complete coverage majority of the time. The configuration of the agents on the graph at some instants of this simulation are provided in Fig. 27.

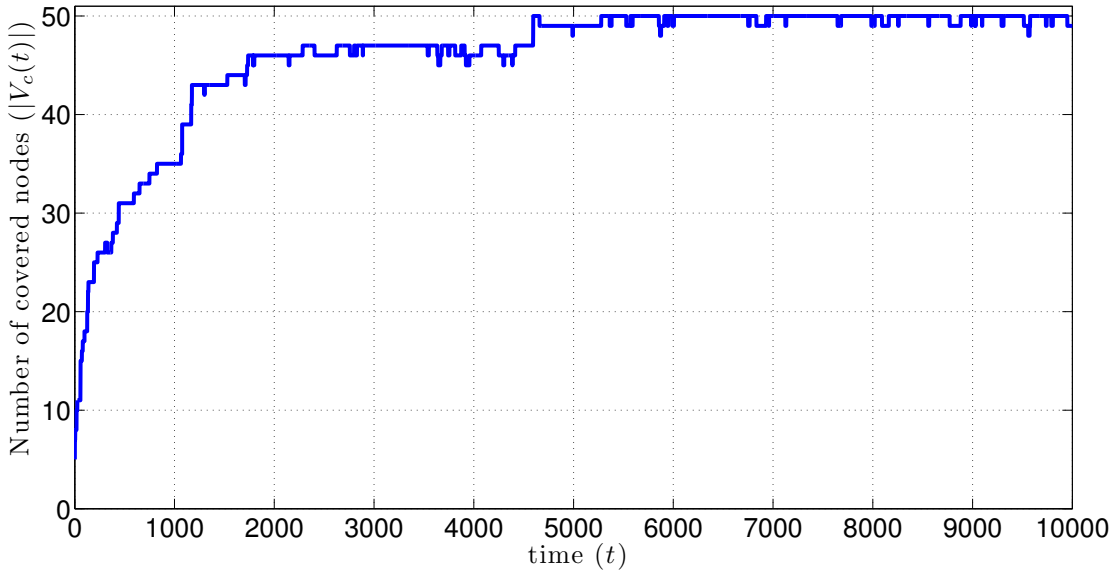


Figure 24. The number of covered nodes as a function of time. 13 homogeneous agents initially start at an arbitrary location and use the proposed method to cover a graph consisting of 50 nodes. The number of covered nodes is initially 5, whereas a complete coverage is maintained with a very high probability after a sufficient amount of time.

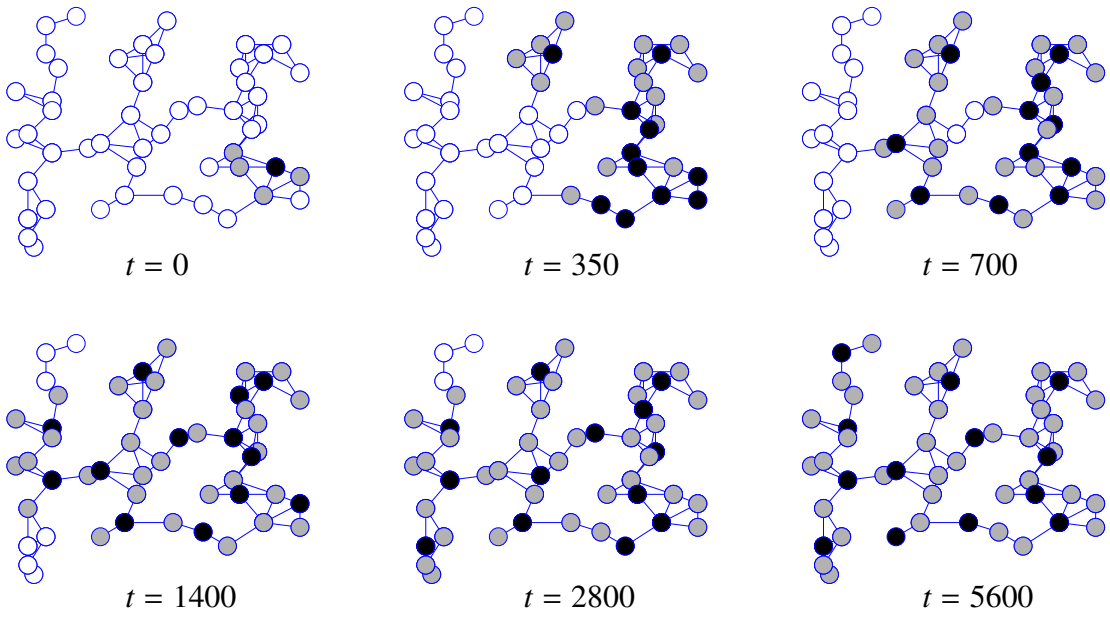


Figure 25. The configuration of the agents on the graph at some instants of the first simulation with 10 homogeneous agents. Each agent has a cover range of 1 and a communication ranges of 3. The nodes having at least one agent located on them are black, the nodes covered by at least one agent are gray, and the nodes that are not covered are white.

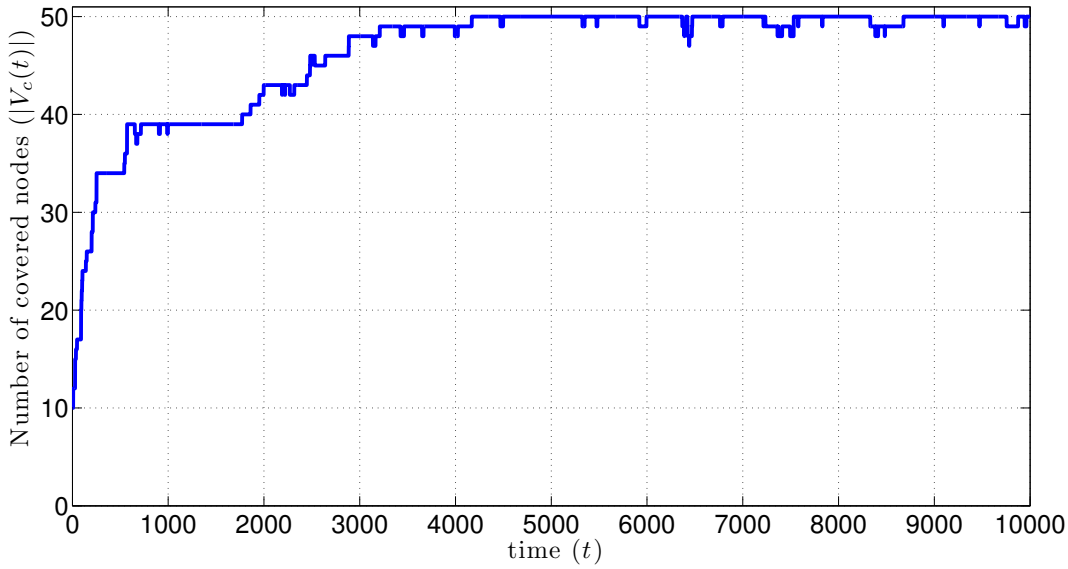


Figure 26. The number of covered nodes as a function of time. 10 heterogeneous agents initially start at an arbitrary location and use the proposed method to cover a graph consisting of 50 nodes. The number of covered nodes is initially 10, whereas a complete coverage is maintained with a very high probability after a sufficient amount of time.

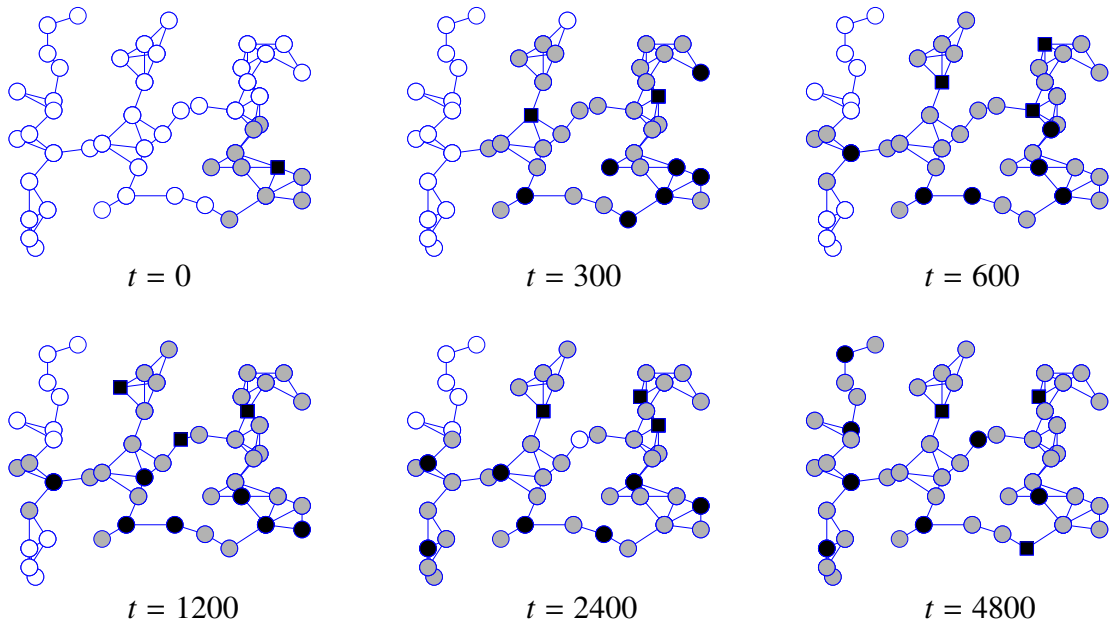


Figure 27. The configuration of the agents on the graph at some instants of the first simulation with 10 heterogeneous agents.Seven agents have cover ranges of 1 and communication ranges of 4, whereas the remaining three agents have cover ranges of 2 and communication ranges of 5. The nodes having at least one agent located on them are black (square if at least one of the agents on it has a cover range of 2), the nodes covered by at least one agent are gray, and the nodes that are not covered are white.

CHAPTER 6

COMMUNICATION-FREE GRAPH COVERAGE

This chapter presents a communication-free solution for optimal protection of networks by a group of mobile security resources. To this end, the game theoretic approach in the previous chapter is extended to drive the resources (agents) to optimal locations on the network without relying on any explicit communications among the agents. In particular, a communication-free algorithm is designed for the agents to follow in a repetitive play of Γ_{DGC} that was designed in the previous chapter. In a communication-free setting, the agents can not compute the exact utilities resulting from their actions. Hence, they need to operate based on some estimated utilities. In the proposed method, an updating agent obtains these estimations by moving around its current coverage area to see which of those nodes (if any) are also covered by some other agents. Using the proposed method, the agents can asymptotically maintain globally optimum coverage with an arbitrarily high probability.

6.1 Problem Formulation

In the previous chapter, the distributed graph coverage (DGC) problem was solved for a setting, where each agent $i \in I$ has a cover range, δ_i, and a communication range, δ_i^c. As such, each agent can cover all the nodes within δ_i, and broadcasts the list of nodes covered by itself to the other agents within its communication range, δ_i^c. A game theoretic solution was proposed for this setting by designing a corresponding potential game, Γ_{DGC}, and employing a learning algorithm, BLLL. Furthermore, Lemma 5.5 provided a sufficient condition for the cover ranges to ensure that the agents can compute the utilities needed to execute the BLLL algorithm. In Theorem 5.6, it was shown that if the agents with such sufficiently large communication ranges follow the BLLL algorithm in a repetitive play of Γ_{DGC}, then the stochastically stable action profiles are the coverage maximizers . As such, the agents asymptotically maintain maximum coverage with an arbitrarily large probability.

The BLLL algorithm requires the agents to be able to measure their utilities resulting from their current actions as well as the hypothetical utilities they may gather by unilaterally switching to some other actions. Alternatively, payoff-based algorithms may be utilized to avoid the necessity to compute those hypothetical utilities. If the agents execute a payoff-based algorithm in Γ_{DGC}, then they only need to compute their current utilities, and the sufficient communication ranges provided in Lemma 5.5 can be reduced by one. A payoff-based implementation of BLLL was presented in [85]. Alternatively, a payoff-based learning algorithm to achieve power-aware coverage on a discretized space was presented in [57]. Note that any algorithm, whether it is payoff-based or not, requires some communications in Γ_{DGC} if the agents need to know their exact utilities to execute the algorithm. Note that agents with overlapping coverage are not necessarily withing the sensing range of each other. As such, the agents cannot measure the number of nodes that are only covered by themselves in a communication-free setting. However, if all the agents have identical sensing ranges, i.e. $\delta_1 = \delta_2 = \ldots = \delta_m = \delta$, then each agent detect if any other agent is also covering its current position as illustrated in Fig. 28.

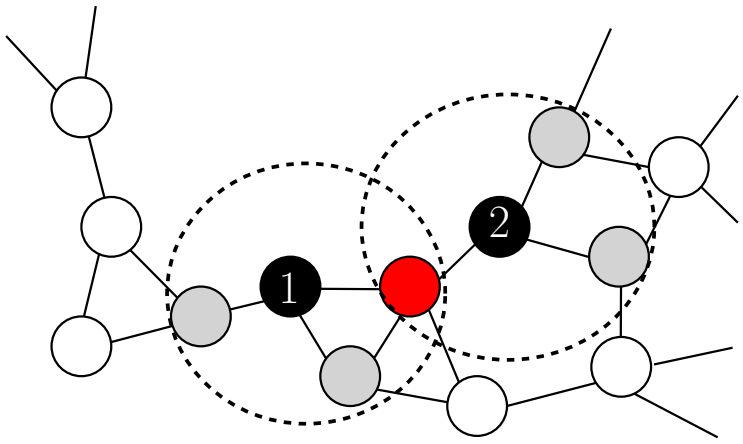


Figure 28. Distributed graph coverage by agents with identical sensing ranges and no communication capabilities. Agents 1 and 2 do not know the node in red is covered by both of them. However, each of them knows its current position is covered only by itself since no other agent is within its sensing range.

Since the exact utilities in Γ_{DGC} are not measurable in a communication-free setting, the agents need to update their actions based on some estimated utilities. This chapter presents

a such communication-free coverage maximization algorithm.

6.2 Proposed Solution

This section presents the proposed communication-free coverage maximization algorithm (CFCM) to be followed by agents with identical sensing ranges in a repetitive play of Γ_{DGC}. Without communications, the agents cannot compute their exact utilities at each time step. However, if all the agents have identical sensing ranges, i.e. $\delta_1 = \delta_2 = \ldots = \delta_m = \delta$, then each agent, i, observes a partial utility representing whether its current position is covered by any other agent. Let $u_i(a_i, a_{-i})$ be the binary partial utility denoting if agent i is the only agent covering its current position, i.e.

$$u_i(a_i, a_{-i}) = \begin{cases} 1 & \text{if } d(a_i, a_j) > \delta \quad \forall j \neq i, \\ 0 & \text{otherwise.} \end{cases} \tag{71}$$

In the proposed algorithm, the agents update their actions based on estimated utilities gathered by moving around to sample the partial utilities from the nodes within their sensing range, δ. Note that, for Γ_{DGC}, the agent utilities in (58) can be expanded as

$$U_i(a) = |\mathcal{N}_{a_i}^{\delta} \setminus \bigcup_{j \neq i} \mathcal{N}_{a_j}^{\delta}| = \sum_{v \in \mathcal{N}_{a_i}^{\delta}} u_i(v, a_{-i}), \tag{72}$$

where, each $u_i(v, a_{-i})$ is the partial utility agent i gathers by covering node v. Note that $U_i(a)$ is not available to i in the communication-free setting. However, assuming that the nearby agents will be stationary for a sufficient amount of time, i can construct an estimate by visiting each $v \in \mathcal{N}_{a_i}^{\delta}$ and sampling $u_i(v, a_{-i})$ as defined in (71). Note that the resulting estimation is not necessarily equal to the actual utility since multiple agents may be moving simultaneously as illustrated in Fig. 29. However, if the probability of multiple agents moving at the same time is sufficiently small, then a noisy best-response based on the estimated utilities can provide an asymptotic performance similar to the solution in the previous chapter. The proposed communication-free algorithm is based on this approach.

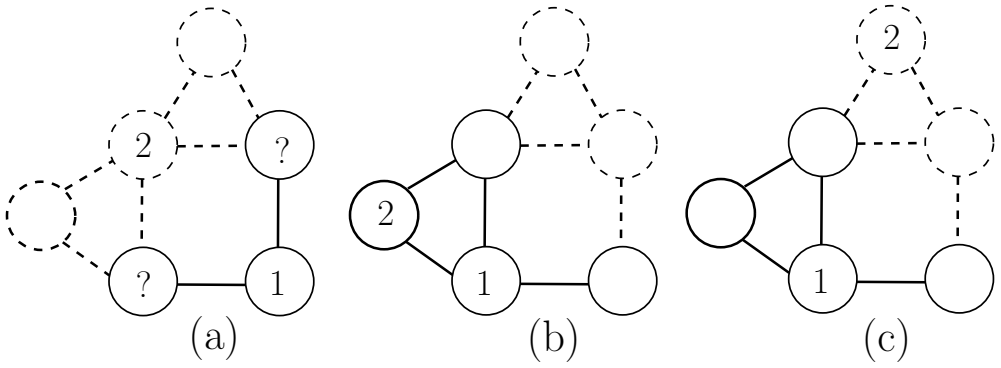


Figure 29. Two agents with sensing ranges of 1 are located on a graph as in (a). Part of the graph that is not sensed by agent 1 is dashed in the figures. Agent 1 can estimate its utility form the action profile in (a) by sampling the partial utilities from the nodes in its sensing range. If agent 2 is stationary in the meantime, then the resulting estimation will be true. However, if agent 2 is also moving, then the sampled partial utilities may be true as in (b) or false as in (c).

In the proposed algorithm, each agent i is either stationary or experimenting. Stationary agents are the ones that maintain their actions in the next time step with a high probability, $1 - \epsilon^r$, where $\epsilon \in \mathfrak{R}^+$ is the noise parameter (small), and $r \in \mathfrak{R}^+$ sets the likelihood (more likely for higher r) of having a single agent experimenting at a time. Each stationary agent starts an experiment, with probability ϵ^r, to compare its current action, a_i^1, to an alternative randomly picked from its constrained action set, $a_i^2 \in A_i^c(a_i^1)$, where $A_i^c(a_i^1)$ the local neighborhood as given in (63). In this aspect, the agent behavior is similar to the payoff-based BLLL in [85]. However, since the agents can only observe some partial utilities, $u_i(a_i, a_{-i})$, an experiment consists of visiting all the nodes in $\mathcal{N}^{\delta}_{a_i^1} \cup \mathcal{N}^{\delta}_{a_i^2}$ to see which of those nodes are also covered by some other agents. During an experiment, the corresponding utility estimations, $\hat{U}_i^1$ and $\hat{U}_i^2$, are obtained by combining the sampled partial utilities.

In order to compare two actions, a_i^1 and $a_i^2 \in A_i^c(a_i^1)$, agent i follows an experiment path between a_i^1 and a_i^2. An experiment path is a finite path from a_i^1 to a_i^2 that traverses all the nodes in $\mathcal{N}^{\delta}_{a_i^1} \cup \mathcal{N}^{\delta}_{a_i^2}$.

Definition *(Experiment Path)*: Let δ be the sensing range of the agents. For any a_i^1 and $a_i^2 \in A_i^c(a_i^1)$, a finite path, $\{a_i^1, \ldots, a_i^2\}$, is an experiment path if it traverses $\mathcal{N}_{a_i^1}^{\delta} \cup \mathcal{N}_{a_i^2}^{\delta}$.

For any a_i^1 and $a_i^2 \in A_i^c(a_i^1)$, an experiment path can be obtained locally by utilizing methods such as depth-first search or breadth-first search (e.g., [87]). In the CFCM algorithm, $\mathcal{E}(a_i^1, a_i^2)$ denotes an experiment path between a_i^1 and a_i^2. During the experiment, the agent traverses its experiment path to obtain the estimated utilities, $\hat{U}_i^1$ and $\hat{U}_i^2$. For simplicity, a partial utility from a node is sampled only at the last visit to that node during the experiment. As such, if it is the last visit of the current position, a_i, and the agent does not sense any other agent within δ, then the utility estimations for the candidate actions within δ are incremented by 1. Once the experiment path is traversed, the agent randomly chooses between the two candidate actions based on the estimated utilities, $\hat{U}_i^1$ and $\hat{U}_i^2$. At the next time step, the agent becomes stationary at its chosen action until its next experiment.

For the CFCM algorithm, the state of any agent i can be defined as

$$x_i = [\ S_i \quad k_i \quad \hat{U}_i^1 \quad \hat{U}_i^2\ ], \tag{73}$$

where S_i is a sequence of actions, which is either a singleton (stationary) or an experiment path (experimenting), $k_i \in \{1, \ldots, |S_i|\}$ is an index variable denoting which action in S_i is currently taken by the agent, and $\hat{U}_i^1, \hat{U}_i^2$ are the estimations for $U_i(a_i^1, a_{-i})$ and $U_i(a_i^2, a_{-i})$, respectively. In this representation, the current action, a_i, and the candidate actions, a_i^1 and a_i^2, are given by S_i and k_i as

$$a_i = S_i(k_i), \tag{74}$$

$$a_i^1 = S_i(1), \tag{75}$$

$$a_i^2 = S_i(|S_i|), \tag{76}$$

where $S_i(k_i)$ denotes the k_i^{th} element in S_i, and $|S_i|$ denotes the length of S_i.

Algorithm VI: Communication-free Coverage Maximization (CFCM)

1 : **initialization:** $a_i \in A_i$ arbitrary, $S_i = \{a_i\}$, $k_i = 1$, $\hat{U}_i^1 = \hat{U}_i^2 = 0$

$\epsilon \in \Re^+$ (small), $r \in \Re^+$.

2 : **repeat**

3 : $a_i = S_i(k_i)$, $a_i^1 = S_i(1)$, $a_i^2 = S_i(|S_i|)$.

4 : **if** $(|S_i| = 1)$

5 : Generate a random (uniform) $\gamma \in [0, 1]$.

6 : **if** $(\gamma \leq \epsilon^r)$

7 : a_i^2 is randomly (uniform) chosen over $A_i^c(a_i^1)$.

8 : $S_i = \mathcal{E}(a_i^1, a_i^2)$.

9 : **end if**

10 : **else**

11 : **if** $(k_i \geq k, \ \forall k \in \{k \mid S_i(k) = a_i\})$

12 : $\hat{U}_i^1 = \hat{U}_i^1 + u_i(a_i, a_{-i})$, if $a_i \in \mathcal{N}_{a_i^1}^{\delta}$.

13 : $\hat{U}_i^2 = \hat{U}_i^2 + u_i(a_i, a_{-i})$, if $a_i \in \mathcal{N}_{a_i^2}^{\delta}$.

14 : **end if**

15 : **if** $(k_i = |S_i|)$

16 : $\alpha = \epsilon^{-\hat{U}_i^1}, \beta = \epsilon^{-\hat{U}_i^2}$.

17 : $$S_i = \begin{cases} \{a_i^1\} & \text{w.p. } \frac{\alpha}{\alpha+\beta}, \\ \{a_i^2\} & \text{otherwise.} \end{cases}$$

18 : $k_i = 1$, $\hat{U}_i^1 = \hat{U}_i^2 = 0$.

19 : **else**

20 : $k_i = k_i + 1$.

21 : **end if**

22 : **end if**

23 : **end repeat**

Note that the CFCM algorithm is memoryless since the state of every agent in the next time step is independent of the past trajectory. As such, if all agents follow the CFCM algorithm, then a Markov chain is induced over the state space, $\mathcal{X}$, where each $x \in \mathcal{X}$ is the global state obtained by concatenating the states of all agents, i.e.

$$x = [x_1, x_2, \ldots, x_m], \tag{77}$$

where each x_i denotes the state of the corresponding agent as given in (73).

In the remainder of this section, it will be shown that the Markov chain induced by the CFCM algorithm is a regular perturbed Markov chain. Furthermore, it will be shown that the stochastically stable states are the coverage maximizers for sufficiently large values of the parameter r. To this end, a resistance-tree analysis will be presented. Prior to this analysis, some preliminaries about resistance trees are provided.

6.2.1 Stochastic Stability and Resistance Trees

Let P_ϵ be a regular perturbed Markov chain over a state space $\mathcal{X}$. For any $x \in \mathcal{X}$, a spanning tree rooted at x, $\mathcal{T}_x$, is a directed graph, where the nodes correspond to states, directed edges correspond to some feasible state transitions, and there is a unique directed path on $\mathcal{T}_x$ from any state $x' \neq x$ to x. The resistance of such a tree, $R(\mathcal{T}_x)$, is defined as the sum of the resistances of its edges, where the resistance of each edge is given as in (54). $\mathcal{T}_x^*$ is called a minimum resistance tree if $R(\mathcal{T}_x^*) \leq R(\mathcal{T}_x)$ for any $\mathcal{T}_x$, i.e. any spanning tree rooted at x has at least as much resistance as $\mathcal{T}_x^*$. The stochastic potential of a state, x, is defined as the total resistance of its minimum resistance tree, $R(\mathcal{T}_x^*)$.

Lemma 6.1 *[88] Let P_ϵ be a regular perturbed Markov chain. Any $x \in \mathcal{X}$ is stochastically stable if and only if x is a recurrent state of the unperturbed chain, P_0, with the minimum stochastic potential.*

In light of Lemma 6.1 and the definition of stochastic potential, the stochastically stable states can be determined through a resistance tree analysis.

6.2.2 Limiting Behavior

For any $x \in \mathcal{X}$, the agents can be grouped into two distinct sets consisting of the stationary agents, $I_s(x)$, and the experimenting agents, $I_e(x)$, as

$$I_s(x) = \{i \in I \mid |S_i| = 1\}, \tag{78}$$

$$I_e(x) = I \setminus I_s(x). \tag{79}$$

Using these sets, for any feasible transition, $x \to x+$, the agents can be grouped into 4 disjoint sets based on the transition of their individual states:

$$I_{ss}(x, x^+) = I_s(x) \cap I_s(x^+), \tag{80}$$

$$I_{se}(x, x^+) = I_s(x) \cap I_e(x^+), \tag{81}$$

$$I_{ee}(x, x^+) = I_e(x) \cap I_e(x^+), \tag{82}$$

$$I_{es}(x, x^+) = I_e(x) \cap I_s(x^+), \tag{83}$$

where $I_{ss}(x, x^+)$ are the agents that remain stationary, $I_{se}(x, x^+)$ are the ones starting to experiment, $I_{ee}(x)$ are the experimenting agents that have not completed moving along their experiment paths, and $I_{es}(x)$ are the agents that have completed traversing their experiment paths and choose between their candidate actions.

The agents in $I_{es}(x, x^+)$ can be further partitioned as the ones choosing their first candidate action and the ones that choose their second candidate action, i.e.

$$I^1_{es}(x, x^+) = \{i \in I_{es}(x, x^+) \mid a_i^+ = a_i^1\}, \tag{84}$$

$$I^2_{es}(x, x^+) = \{i \in I_{es}(x, x^+) \mid a_i^+ = a_i^2\}. \tag{85}$$

Note that the agents in $I_{es}(x, x^+)$ do not necessarily choose the action resulting in the higher estimated utility. For each $i \in I$, let $\hat{U}_i^*$=max$\{\hat{U}_i^1, \hat{U}_i^2\}$. Then, the amount of estimated utility that is denied in the transition $x \to x^+$ is given as

$$\Delta_i(x_i, x_i^+) = \begin{cases} \hat{U}_i^* - \hat{U}_i^1 & \text{if } i \in I^1_{es}(x, x^+), \\ \hat{U}_i^* - \hat{U}_i^2 & \text{if } i \in I^2_{es}(x, x^+), \\ 0 & \text{otherwise.} \end{cases} \tag{86}$$

Lemma 6.2 *Let all agents follow the CFCM algorithm to cover a graph, $\mathcal{G} = (V, E)$, and let Δ^* be*

$$\Delta^* = \max_{(v,v')\in E} |\mathcal{N}_v^\delta \setminus \mathcal{N}_{v'}^\delta|. \tag{87}$$

Then, for any feasible transition $x \to x^+$,

$$\Delta^* \geq \max_{i\in I} \Delta_i(x_i, x_i^+). \tag{88}$$

Proof: Let $x \to x^+$ be a feasible transition. For any $i \in I_s(x)$, $\hat{U}_i^1 = \hat{U}_i^2 = 0$. On the other hand, for any $i \in I_e(x)$, the sampled partial utilities from the nodes $\mathcal{N}_{a_i^1}^\delta \cap \mathcal{N}_{a_i^2}^\delta$, contribute equally to both $\hat{U}_i^1$ and $\hat{U}_i^2$. Hence,

$$\max\{|\mathcal{N}_{a_i^1}^\delta \setminus \mathcal{N}_{a_i^2}^\delta|, |\mathcal{N}_{a_i^2}^\delta \setminus \mathcal{N}_{a_i^1}^\delta|\} \geq |\hat{U}_i^1 - \hat{U}_i^2|, \ \forall i \in I. \tag{89}$$

In light of (89) and (86),

$$\max\{|\mathcal{N}_{a_i^1}^\delta \setminus \mathcal{N}_{a_i^2}^\delta|, |\mathcal{N}_{a_i^2}^\delta \setminus \mathcal{N}_{a_i^1}^\delta|\} \geq \Delta_i(x_i, x_i^+), \ \forall i \in I. \tag{90}$$

Since $(a_i^1, a_i^2) \in E$ for any $i \in I_e(x)$, (87) implies

$$\Delta^* \geq \max\{|\mathcal{N}_{a_i^1}^\delta \setminus \mathcal{N}_{a_i^2}^\delta|, |\mathcal{N}_{a_i^2}^\delta \setminus \mathcal{N}_{a_i^1}^\delta|\}, \ \forall i \in I. \tag{91}$$

Finally, (90) and (91) together imply (88).

■

Next, it will be shown that the perturbed Markov process induced by the CFCM algorithm is a regular perturbed process. Furthermore, if the agents follow the CFCM with $r > \Delta^*$, then the paths between all-stationary ($I_s(x) = I$) states on any minimum resistance tree consist of unilateral experimentations. Note that with unilateral experimentations, the utility estimations at the end of an experimentation path is equal to the true utilities. As such, for $r > \Delta^*$, the stochastically stable states are the all-stationary states maximizing the number of covered nodes.

Lemma 6.3 *If all agents employ the CFCM algorithm, then a regular perturbed Markov chain is induced over $\mathcal{X}$, and the resistance of any feasible transition, $x \to x^+$, is*

$$R(x, x^+) = r|I_{se}(x, x^+)| + \sum_{i \in I_{es}(x,x^+)} \Delta_i(x_i, x_i^+). \tag{92}$$

Proof: In P_ϵ, for $\epsilon > 0$, any all-stationary state can be reached from any other all-stationary state by a sequence of experiments, given the graph is connected. Furthermore, any state that is not all-stationary lies on a feasible path between two all-stationary states. Hence, P_ϵ is irreducible. Furthermore, since the stationary agents remain stationary with probability $1 - \epsilon^r$, aperiodicity immediately follows from the resulting self-loops at all-stationary states.

The probability any feasible transition from x to x^+, given in P_ϵ, is the joint probability of state transitions of individual agents. Note that for any agent, $i \in I_{ee}(x)$, the transition from x_i to x_i^+ does not have any randomness. Hence, the probability of transition from x to x^+ is

$$P_\epsilon(x, x^+) = \Pr[I_{es}^1(x, x^+)] \Pr[I_{es}^2(x, x^+)] \Pr[I_{ss}(x, x^+)] \Pr[I_{se}(x, x^+)], \tag{93}$$

where each term on the right side of (93) denote the joint probability of state transitions for the agents in the corresponding subset, and they are given as

$$\Pr[I_{es}^1(x, x^+)] = \prod_{i \in I_{es}^1(x,x^+)} \frac{\epsilon^{-\hat{U}_i^1}}{\epsilon^{-\hat{U}_i^1} + \epsilon^{-\hat{U}_i^2}}, \tag{94}$$

$$\Pr[I_{es}^2(x, x^+)] = \prod_{i \in I_{es}^2(x,x^+)} \frac{\epsilon^{-\hat{U}_i^2}}{\epsilon^{-\hat{U}_i^1} + \epsilon^{-\hat{U}_i^2}}, \tag{95}$$

$$\Pr[I_{ss}(x, x^+)] = \prod_{i \in I_{ss}(x,x^+)} (1 - \epsilon^r), \tag{96}$$

$$\Pr[I_{se}(x, x^+)] = \prod_{i \in I_{se}(x,x^+)} \frac{\epsilon^r}{|A_i^c(a_i^1)|} \Pr[S_i^+; a_i^1, a_i^2], \tag{97}$$

where $\Pr[S_i^+; a_i^1, a_i^2]$ is the probability of having S_i^+ as the experiment path for an agent comparing a_i^1 and a_i^2. $\Pr[S_i^+; a_i^1, a_i^2]$ depends on the function $\mathcal{E}(a_i^1, a_i^2)$, and it is independent

of ϵ. Plugging (94)-(97) into (93), one can verify that the resistance $R(x, x^+)$ given in (92) satisfies

$$0 < \lim_{\epsilon \to 0^+} \frac{P_\epsilon(x, x^+)}{\epsilon^{R(x,x^+)}} < \infty. \tag{98}$$

■

Since CFCM induces a regular perturbation, only the recurrent states of the unperturbed ($\epsilon = 0$) chain can be stochastically stable as given in Lemma 6.1. Note that if $\epsilon = 0$, then no agent starts an experiment. In that case, the set of recurrent states, $\mathcal{X}_R^0$, contain the all-stationary states whereas all the remaining states form the set of transient states, $\mathcal{X}_T^0$, i.e.

$$\mathcal{X}_R^0 = \{x \mid I_s(x) = I\}, \tag{99}$$

$$\mathcal{X}_T^0 = \mathcal{X} \setminus \mathcal{X}_R^0. \tag{100}$$

Definition *(Unilateral Experimentation Path)*: A feasible sequence of states, $\mathcal{P} = \{x^1, x^2, \dots x^n\}$, is a unilateral experimentation path if $x^1, x^n \in \mathcal{X}_R^0$, $x^2, \dots, x^{n-1} \in \mathcal{X}_T^0$ and for all $1 \leq p \leq n-1$

$$|I_{se}(x^p, x^{p+1})| = \begin{cases} 1 & \text{if } p = 1, \\ 0 & \text{otherwise.} \end{cases} \tag{101}$$

Lemma 6.4 *Let $\mathcal{T}^*$ be a minimum resistance tree, and let $x \to x^+ \in \mathcal{T}^*$. If $x \in \mathcal{X}_R^0$, then $|I_{se}(x, x^+)| = 1$.*

Proof: Since $x \in \mathcal{X}_R^0$, $|I_{se}(x, x^+)| > 0$, as otherwise, $x^+ = x$ and $x \to x^+$ cannot be contained in a tree. Assume that $|I_{se}(x, x^+)| > 1$. Then, choose an arbitrary $i \in I_{se}(x, x^+)$ to define an $\tilde{x}^+ \neq x$ as

$$\tilde{x}_j^+ = \begin{cases} x_j^+ & \text{if } j \neq i, \\ x_i & \text{otherwise.} \end{cases} \tag{102}$$

Note that $x \to \tilde{x}^+$ is a feasible transition, and $R(x, \tilde{x}^+) = R(x, x^+) - r(|I_e(x, x^+)| - 1)$. Replacing $x \to x^+$ with $x \to \tilde{x}^+$ would give an alternative tree with a smaller resistance, which contradicts with $\mathcal{T}$ being a minimum resistance tree. ■

Lemma 6.5 *Let $\mathcal{T}^*$ be a minimum resistance tree, and let $x \to x^+ \in \mathcal{T}^*$. If $x \in \mathcal{X}_T^0$ and $r > \Delta^*$, then we have $|I_{se}(x, x^+)| < |I_e(x)|$.*

Proof: Since $x \in \mathcal{X}_T^0$, $I_{se}(x, \tilde{x}^+) = \emptyset$ doesn't imply $\tilde{x}^+ = x$. Hence, there exists an $\tilde{x}^+ \neq x$ such that $x \to \tilde{x}^+$ is feasible and $I_{se}(x, \tilde{x}^+) = \emptyset$. For any such $\tilde{x}^+$, we have

$$R(x, \tilde{x}^+) - R(x, x^+) \leq -r|I_{se}(x, x^+)| + |I_{es}(x, \tilde{x}^+)|\Delta^*. \tag{103}$$

Note that $|I_{es}(x, \tilde{x}^+)| \leq |I_e(x)|$. Hence, given $r > \Delta^*$, the right side of (103) is negative for any $|I_{se}(x, x^+)| \geq |I_e(x)|$. In that case, replacing $x \to x^+$ with $x \to \tilde{x}^+$ would give an alternative tree with a smaller resistance, which contradicts with $\mathcal{T}$ being a minimum resistance tree. Consequently, $|I_{se}(x, x^+)| < |I_e(x)|$. ■

Lemma 6.6 *Let $r > \Delta^*$, and let $\mathcal{P} = \{x^1, x^2, \dots x^n\}$ be a sequence of states, where $x^1, x^n \in \mathcal{X}_R^0$ and $x^2, \dots, x^{n-1} \in \mathcal{X}_T^0$. If $\mathcal{P} \in \mathcal{T}$ for some minimum resistance tree $\mathcal{T}$, then $\mathcal{P}$ is a unilateral experimentation path.*

Proof: Since $x^1 \in \mathcal{X}_R^0$, from Lemma 6.4, we have $|I_{se}(x^1, x^2)| = 1$ leading to $|I_e(x^2)| = 1$. Furthermore, for $r > \Delta^*$, from Lemma 6.5, we have $|I_{se}(x^2, x^3)| = 0$. Hence, we have $|I_e(x^3)| \leq 1$. Using Lemma 6.5 recursively along $\mathcal{P}$ we obtain

$$|I_{se}(x^p, x^{p+1})| = \begin{cases} 1 & \text{if } p = 1, \\ 0 & \text{otherwise.} \end{cases} \tag{104}$$

Hence, $\mathcal{P}$ is a unilateral experimentation path. ■

Lemma 6.7 *If $\mathcal{P} = \{x^1, x^2, \dots x^n\}$ be a unilateral experimentation path, then*

$$R(\mathcal{P}) = \sum_{p=1}^{n-1} R(x^p, x^{p+1}) = r + max\{\phi(x^n), \phi(x^1)\} - \phi(x^n). \tag{105}$$

Proof: Since $\mathcal{P} = \{x^1, x^2, \dots x^n\}$ be a unilateral experimentation path, for $x^p, x^{p+1} \in \mathcal{X}_T^0$, we have

$$|I_{se}(x^p, x^{p+1})| = |I_{es}(x^p, x^{p+1})| = 0. \tag{106}$$

Hence, such transitions have zero resistance, resulting in

$$R(\mathcal{P}) = R(x^1, x^2) + R(x^{n-1}, x^n). \tag{107}$$

Note that, since $x^1 \in \mathcal{X}_R^0$ and $\mathcal{P}$ is a unilateral experimentation path, we have $R(x^1, x^2) = r$ and $R(x^{n-1}, x^n) = \Delta_i(x_i^{n-1}, x_i^n)$, where $i \in I$ is the unique experimenting agent. Since all the other agents are stationary, i.e. a_{-i} is constant along $\mathcal{P}$, the estimated utilities satisfy

$$(\hat{U}_i^1)^{n-1} = \sum_{v \in \mathcal{N}_{a_i^1}^{\delta}} u(v, a_{-i}) = U_i(a_i^1, a_{-i}), \tag{108}$$

$$(\hat{U}_i^2)^{n-1} = \sum_{v \in \mathcal{N}_{a_i^2}^{\delta}} u(v, a_{-i}) = U_i(a_i^2, a_{-i}). \tag{109}$$

Plugging (108) and (109) into (86) we obtain

$$\Delta_i(x_i^{n-1}, x_i^n) = max\{U_i(x^n), U_i(x^1)\} - U_i(x^n). \tag{110}$$

Since Γ_{DGC} is a potential game, in light of (53), from (110) we obtain

$$\Delta_i(x_i^{n-1}, x_i^n) = max\{\phi(x^n), \phi(x^1)\} - \phi(x^n). \tag{111}$$

■

Lemma 6.8 *Let $r > \Delta^*$, and let $\mathcal{T}_x^*$ and $\mathcal{T}_{x'}^*$ be minimum resistance trees rooted at some $x, x' \in \mathcal{X}_R^0$. Then,*

$$R(\mathcal{T}_x^*) \leq R(\mathcal{T}_{x'}^*) \Rightarrow \phi(x) \geq \phi(x'). \tag{112}$$

Proof: For $r > \Delta^*$, in light of Lemma 6.6, paths between states in $\mathcal{X}_R^0$ on a minimum resistance tree consists of unilateral experimentations. Let $x_R^0 \in \mathcal{X}_R^0$, and let $\mathcal{T}_{x_R^0}^*$ be a minimum resistance tree rooted at x_R^0. Let $x_R^n \in \mathcal{X}_R^0$ be a state such that $R(\mathcal{T}_{x_R^0}^*) \leq R(\mathcal{T}_{x_R^n}^*)$ and the unique path, $\mathcal{P} \in \mathcal{T}_{x_R^0}^*$, from x_R^n to x_R^0 consists of n unilateral experimentations, i.e.

$$R(\mathcal{P}) = \sum_{k=1}^{n} R(\mathcal{P}_k), \tag{113}$$

where $\mathcal{P}_k$ is the unilateral experimentation starting at x_R^{n-k+1} and ending at x_R^{n-k}. Note that, for each such $\mathcal{P}_k$, there exists a feasible unilateral experimentation path $\mathcal{P}'_k$ in the reversed direction, starting at x_R^{n-k} and ending at x_R^{n-k+1}. Replacing each $\mathcal{P}_k$ with $\mathcal{P}'_k$, one can construct a tree rooted at $\mathcal{T}_{x_R^n}$. Note that the resistances of these trees satisfy

$$\begin{aligned} R(\mathcal{T}_{x_R^n}) - R(\mathcal{T}^*_{x_R^0}) &= \sum_{k=1}^{n} (R(\mathcal{P}'_k) - R(\mathcal{P}_k)) \\ &= \sum_{k=1}^{n} (\phi(x_R^{n-k}) - \phi(x_R^{n-k+1})) \\ &= \phi(x_R^0) - \phi(x_R^n). \end{aligned} \tag{114}$$

Note that by definition $R(\mathcal{T}^*_{x_R^n}) \leq R(\mathcal{T}_{x_R^n})$. Hence, if $R(\mathcal{T}^*_{x_R^0}) \leq R(\mathcal{T}^*_{x_R^n})$, then $R(\mathcal{T}^*_{x_R^0}) \leq R(\mathcal{T}_{x_R^n})$ for any $\mathcal{T}_{x_R^n}$. Plugging this into (114), we obtain $\phi(x_R^0) \geq \phi(x_R^n)$

■

Theorem 6.9 *Let each agent follow the CFCM algorithm with $r > \Delta^*$, and let x be a stochastically stable state of the resulting Markov chain, P_ϵ. Then, $x \in \mathcal{X}_R^0$ and*

$$|V_c(x)| \geq |V_c(x')|, \ \forall x' \in \mathcal{X}_R^0. \tag{115}$$

Proof: Let x be a stochastically stable state of P_ϵ. Due to Lemma 6.1, $x \in \mathcal{X}_R^0$ and $R(\mathcal{T}^*_x) \leq R(\mathcal{T}^*_{x'})$ for all $x' \in \mathcal{X}_R^0$. In light of Lemma 6.8, if $r > \Delta^*$, then $R(\mathcal{T}^*_x) \leq R(\mathcal{T}^*_{x'})$ implies $\phi(x) \geq \phi(x')$ for all $x' \in \mathcal{X}_R^0$. As such, (115) is satisfied since $\phi(x) = |V_c(x)|$. ■

Theorem 6.9 indicates that if the agents follow the CFCM algorithm with sufficiently large r, then the stochastically stable states are all-stationary states globally maximizing the number of covered nodes. As such, if r is taken sufficiently large, then the agents asymptotically maintain maximum coverage with an arbitrarily high probability for arbitrarily small values of ϵ. Note that $r > \Delta^*$ in Theorem 6.9 is a sufficient condition to ensure the stochastic stability of potential maximizers based on the sufficient unlikeliness of simultaneous experiments as given in Lemma 6.6. However, in many cases, $r > \Delta^*$ may not be necessary since simultaneously updating agents do not spoil the utility estimations of each other as long as they are far from each other on the graph.

6.3 Simulation Results

In this section, some simulation results for the CFCM algorithm are presented. For comparison, the simulation is performed for the same scenario as in the previous chapter. A group of 13 homogeneous agents are initially placed at an arbitrary node of a connected random geometric graph, which consists of 50 nodes and 78 edges, and has a diameter of 17. Each agent has a sensing range of $\delta = 1$, and the agents follow the CFCM algorithm with $\epsilon = 0.015$ and $r = 1.5$. As such, any stationary agent starts an experiment with probability $0.015^{1.5} = 0.0018$.

Initially, the agents cover only 5 nodes. The number of covered nodes throughout a period of 200000 time steps is shown in Fig. 30, whereas the configuration of the agents on the graph at some instants are provided in Fig. 31. As depicted in Fig. 30, after a sufficient amount of time, the agents maintain complete coverage with a very high probability. In particular, for $t \geq 150000$, the average number of covered nodes at each time step is 49.7.

If the results in Figs. 30 and 31. are compared to those in Figs. 24 and 25, it is seen that both the BLLL and the CFCM drive the agents to some global optima in a similar fashion. However, the CFCM algorithm is performing approximately 30 times slower than the BLLL algorithm for this particular scenario and the particular values of ϵ and r. The speed of convergence to the limiting distribution is reduced in the CFCM mostly due to two things: First, BLLL assumes that a randomly picked agent updates its action at each time step, whereas each stationary agent starts an experiment with probability ϵ^r in CFCM. Second, while the exact utilities are available in BLLL, the agents need to gather estimations by moving around for multiple time steps before making any decision in CFCM. Despite the reduced speed of convergence, it should be noted that the main advantage of the CFCM algorithm is to achieve coverage maximization without relying on any communications among the agents. As such, CFCM can be employed to optimally place security resources on networks, even in scenarios that do not allow any explicit communications.

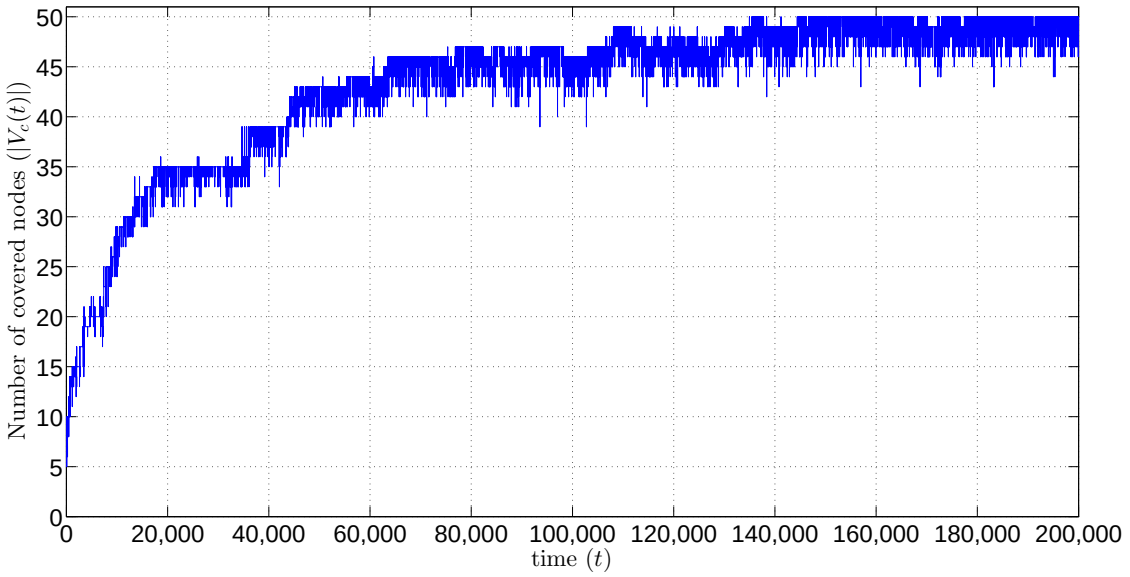


Figure 30. The number of covered nodes as a function of time. Agents initially start at an arbitrary location on a graph consisting of 50 nodes. The number of covered nodes is initially 5, whereas a complete coverage is maintained with a very high probability after a sufficient amount of time.

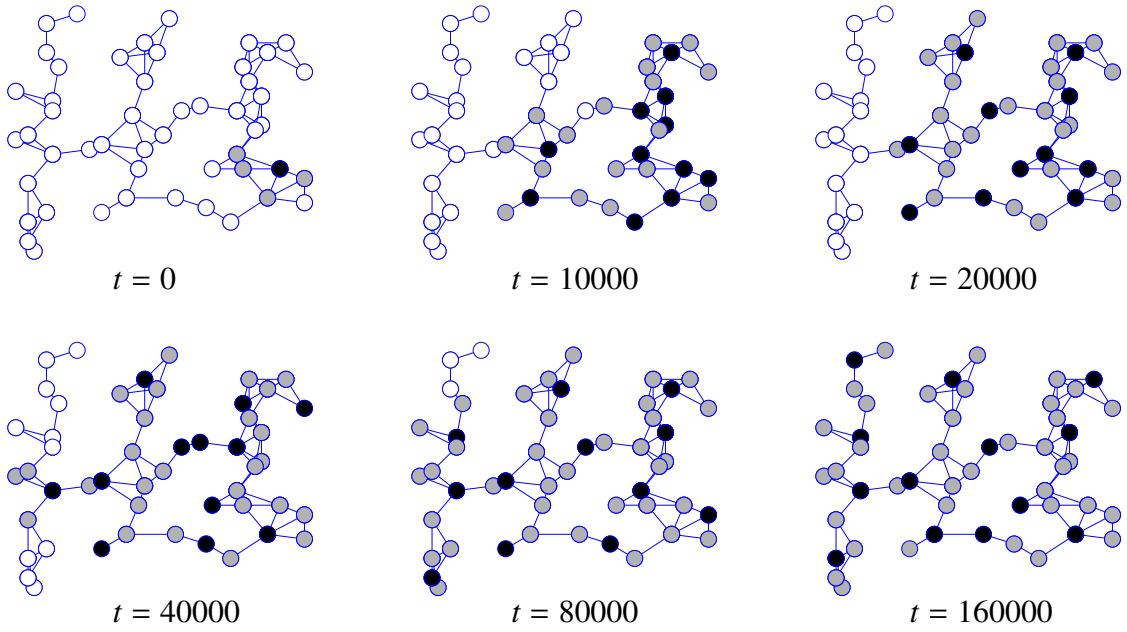


Figure 31. The configuration of the agents on the graph at some instants of the simulation. The nodes having at least one agent located on them are black, the nodes covered by at least one agent are gray, and the nodes that are not covered are white.

CHAPTER 7

CONCLUSIONS

This thesis presents some decentralized graph processes for building robust multi-agent networks through self-organization. Networks are modeled as interaction graphs throughout the thesis, and robustness is pursued through formation of robust interaction graphs and optimal protection of networks by some mobile agents with local capabilities. As such, the thesis consists of two main parts.

The first part of this thesis presents a decentralized graph reconfiguration scheme for formation of robust interaction graphs. The robustness of an interaction graph is quantified via its node/edge expansion ratios. Motivated by the expansion properties of almost every m-regular graph for $m \geq 3$, a decentralized graph reconfiguration scheme is designed to transform any connected interaction graph into a connected random regular graph with a similar sparsity as the initial graph. In order to transform any connected graph into a connected random regular graph, the proposed scheme achieves three global tasks simultaneously while also maintaining the connectivity: balance the degree distribution, randomize the local neighborhoods, and drive the average degree to an integer close to its initial value. The proposed scheme is incrementally built in Chapters 2, 3, and 4. In particular, a graph grammar, Φ^*, is designed and an algorithm for distributed implementation is also provided. In Chapter 2, a single-rule degree regularization grammar, Φ_R, is presented for balancing the degree distribution in a multi-agent network. In Φ_R, if a node has a neighbor with fewer links, then it rewires one of its exclusive neighbors to its neighbor with fewer links. As such, Φ_R maintains the graph connectivity and the total number of edges while minimizing the degree differences in the network. Hence, Φ_R transforms any connected graph into a connected regular graph if the average degree is an integer. This scheme is extended in Chapter 3 to obtain random regular graphs in order to avoid any possible convergence to an undesired regular graph. This is achieved by combining the degree regularization rule

with a neighborhood randomization rule, where adjacent nodes exchange their exclusive neighbors. The resulting grammar, Φ_{RR}, minimizes the degree differences and randomizes the local neighborhoods simultaneously while maintaining the graph connectivity and the total number of edges. As such, it transforms any connected interaction graph with an integer average degree into a connected random regular graph. Note that having an integer average degree is a strong property, and it may not be satisfied by the initial graph in many applications. In order to tackle this issue, Chapter 4 extends Φ_{RR} to obtain random regular graphs even when the initial average degree is not an integer. The resulting grammar, Φ^*, allows for occasional addition and removal of edges in balancing the degree distribution, while ensuring that connectivity is maintained and the total number of edges stays within some proximity of its initial value. In particular, if the initial connected graph has an average degree, k, satisfying $k > 2$, then Φ^* leads to a connected random m-regular graph such that $k \leq m \leq k + 2$. Note that $k > 2$ implies $m \geq 3$, and the graphs observed in the limit are expanders with an arbitrarily high probability for large networks.

The second part of this thesis presents a decentralized scheme for driving a group of arbitrarily deployed mobile security resources to some optimal locations on a network. This problem is essentially a distributed coverage control problem, where the resources with local capabilities are required to optimize their locations to protect as many network components as possible. In order to maximize the number of covered nodes (coverage), regardless of the initial deployment and the graph structure, a solution method should pursue both exploration and exploitation. This thesis presents a such solution by formulating the problem in a game theoretic framework. The proposed solution is presented in Chapters 5 and 6. In Chapter 5, the distributed graph coverage is formulated as a potential game, Γ_{DGC}, such that the coverage is the corresponding potential function. In Γ_{DGC}, the action of a resource (agent) is defined as its position on the graph, and the utility of an agent is the number of nodes that are only covered by itself. It is shown that a group of agents can asymptotically maintain maximum coverage with an arbitrarily high probability by

following a learning algorithm such as binary log-linear learning (BLLL) in a repetitive play of Γ_{DGC}. The method presented in Chapter 5 requires the agents to have sufficient local communications to compute their utilities. Chapter 6 extends the solution approach in Chapter 5 to cases, where no communications are allowed among the agents. In particular, a communication-free coverage maximization algorithm (CFCM) is designed to be followed by agents with identical sensing capabilities in a repetitive play of Γ_{DGC}. Since the agents can not compute their exact utilities in a communication-free setting, any updating agent moves around within its covered area to estimate which of those nodes (if any) are also covered by some other agents. Based on the estimated utilities, the agent randomly chooses one of the two alternative actions. It was shown that the agents can asymptotically maintain maximum coverage with an arbitrarily high probability by following the CFCM algorithm.

REFERENCES

[1] M. Newman, *Networks: an introduction*. Oxford University Press, 2010.

[2] M. O. Jackson, *Social and economic networks*. Princeton University Press, 2010.

[3] O. Mason and M. Verwoerd, "Graph theory and networks in biology," *Systems Biology, IET*, vol. 1, no. 2, pp. 89–119, 2007.

[4] E. Cascetta, *Transportation systems engineering: theory and methods*, vol. 49. Springer, 2001.

[5] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications magazine*, vol. 40, no. 8, pp. 102–114, 2002.

[6] F. Bullo, J. Cortés, and S. Martinez, *Distributed control of robotic networks: a mathematical approach to motion coordination algorithms*. Princeton University Press, 2009.

[7] A. H. Dekker and B. D. Colbert, "Network robustness and graph topology," in *Australasian Conference on Computer Science*, pp. 359–368, 2004.

[8] A. Jamakovic and S. Uhlig, "On the relationship between the algebraic connectivity and graph's robustness to node and link failures," in *EuroNGI Conference on Next Generation Internet Networks*, pp. 96–102, 2007.

[9] R. Olfati-Saber, "Ultrafast consensus in small-world networks," in *American Control Conference*, pp. 2371–2378, 2005.

[10] A. Rahmani, M. Ji, M. Mesbahi, and M. Egerstedt, "Controllability of multi-agent systems from a graph-theoretic perspective," *SIAM Journal on Control and Optimization*, vol. 48, no. 1, pp. 162–186, 2009.

[11] Y.-Y. Liu, J.-J. Slotine, and A.-L. Barabási, "Controllability of complex networks," *Nature*, vol. 473, no. 7346, pp. 167–173, 2011.

[12] H. Ohtsuki, C. Hauert, E. Lieberman, and M. A. Nowak, "A simple rule for the evolution of cooperation on graphs and social networks," *Nature*, vol. 441, no. 7092, pp. 502–505, 2006.

[13] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *Automatic Control, IEEE Transactions on*, vol. 49, no. 9, pp. 1520–1533, 2004.

[14] M. Mesbahi and M. Egerstedt, *Graph theoretic methods in multiagent networks*. Princeton University Press, 2010.

[15] A. Barrat, M. Barthelemy, and A. Vespignani, *Dynamical processes on complex networks*. Cambridge University Press Cambridge, 2008.

[16] V. Latora and M. Marchiori, "Vulnerability and protection of infrastructure networks," *Physical Review E*, vol. 71, no. 1, p. 015103, 2005.

[17] T. T. Kim and H. V. Poor, "Strategic protection against data injection attacks on power grids," *IEEE Transactions on Smart Grid*, vol. 2, no. 2, pp. 326–333, 2011.

[18] V. M. Preciado, M. Zargham, C. Enyioha, A. Jadbabaie, and G. J. Pappas, "Optimal resource allocation for network protection against spreading processes," *IEEE Transactions on Control of Network Systems,*, vol. 1, no. 1, pp. 99–108, 2014.

[19] M. Dziubinski and S. Goyal, "Network design and defence," *Games and Economic Behavior*, vol. 79, pp. 30 – 43, 2013.

[20] H. Jeong, S. P. Mason, A.-L. Barabási, and Z. N. Oltvai, "Lethality and centrality in protein networks," *Nature*, vol. 411, no. 6833, pp. 41–42, 2001.

[21] P. Holme, B. J. Kim, C. N. Yoon, and S. K. Han, "Attack vulnerability of complex networks," *Physical Review E*, vol. 65, no. 5, p. 056109, 2002.

[22] Z. Wang, A. Scaglione, and R. J. Thomas, "Electrical centrality measures for electric power grid vulnerability analysis," in *IEEE Conference on Decision and Control*, pp. 5792–5797, 2010.

[23] S. P. Borgatti, "Centrality and network flow," *Social Networks*, vol. 27, no. 1, pp. 55–71, 2005.

[24] D. Koschützki, K. A. Lehmann, L. Peeters, S. Richter, D. Tenfelde-Podehl, and O. Zlotowski, "Centrality indices," in *Network Analysis*, pp. 16–61, Springer, 2005.

[25] M. Fiedler, "Algebraic connectivity of graphs," *Czechoslovak Mathematical Journal*, vol. 23, no. 2, pp. 298–305, 1973.

[26] D. J. Klein and M. Randić, "Resistance distance," *Journal of Mathematical Chemistry*, vol. 12, no. 1, pp. 81–95, 1993.

[27] M. S. Pinsker, "On the complexity of a concentrator," in *7th International Telegraffic Conference*, vol. 4, pp. 318/1–318/4, 1973.

[28] N. Alon, "Eigenvalues and expanders," *Combinatorica*, vol. 6, no. 2, pp. 83–96, 1986.

[29] B. Mohar, "Isoperimetric numbers of graphs," *Journal of Combinatorial Theory, Series B*, vol. 47, no. 3, pp. 274–291, 1989.

[30] S. Hoory, N. Linial, and A. Wigderson, "Expander graphs and their applications," *Bulletin of the American Mathematical Society*, vol. 43, no. 4, pp. 439–561, 2006.

[31] M. R. Murty, "Ramanujan graphs," *Journal of the Ramanujan Mathematical Society*, vol. 18, no. 1, pp. 33–52, 2003.

[32] J. Friedman, "A proof of alon's second eigenvalue conjecture," in *ACM Symposium on Theory of Computing*, pp. 720–724, 2003.

[33] O. Reingold, S. Vadhan, and A. Wigderson, "Entropy waves, the zig-zag graph product, and new constant-degree expanders," *Annals of Mathematics*, pp. 157–187, 2002.

[34] M. Capalbo, O. Reingold, S. Vadhan, and A. Wigderson, "Randomness conductors and constant-degree lossless expanders," in *ACM Symposium on Theory of Computing*, pp. 659–668, 2002.

[35] E. Rozenman and S. Vadhan, "Derandomized squaring of graphs," in *Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques*, pp. 436–447, Springer, 2005.

[36] M. Morgenstern, "Existence and explicit constructions of q+1 regular ramanujan graphs for every prime power q," *Journal of Combinatorial Theory, Series B*, vol. 62, no. 1, pp. 44–62, 1994.

[37] R. Olfati-Saber, "Algebraic connectivity ratio of ramanujan graphs," in *American Control Conference*, pp. 4619–4624, 2007.

[38] N. C. Wormald, "Models of random regular graphs," *London Mathematical Society Lecture Note Series*, pp. 239–298, 1999.

[39] B. Bollobás, "A probabilistic proof of an asymptotic formula for the number of labelled regular graphs," *European Journal of Combinatorics*, vol. 1, no. 4, pp. 311–316, 1980.

[40] A. Steger and N. C. Wormald, "Generating random regular graphs quickly," *Combinatorics Probability and Computing*, vol. 8, no. 4, pp. 377–396, 1999.

[41] M. Jerrum and A. Sinclair, "Fast uniform generation of regular graphs," *Theoretical Computer Science*, vol. 73, no. 1, pp. 91–100, 1990.

[42] P. Mahlmann and C. Schindelhauer, "Peer-to-peer networks based on random transformations of connected regular undirected graphs," in *ACM Symposium on Parallelism in Algorithms and Architectures*, pp. 155–164, 2005.

[43] C. Law and K.-Y. Siu, "Distributed construction of random expander networks," in *IEEE International Conference on Computer Communications*, pp. 2133–2143, 2003.

[44] C. M. Schneider, A. A. Moreira, J. S. Andrade, S. Havlin, and H. J. Herrmann, "Mitigation of malicious attacks on networks," *Proceedings of the National Academy of Sciences*, vol. 108, no. 10, pp. 3838–3841, 2011.

[45] A. Beygelzimer, G. Grinstein, R. Linsker, and I. Rish, "Improving network robustness by edge modification," *Physica A: Statistical Mechanics and its Applications*, vol. 357, no. 3, pp. 593–612, 2005.

[46] W. Goddard, S. M. Hedetniemi, and S. T. Hedetniemi, "Eternal security in graphs," *J. Combin. Math. Combin. Comput*, vol. 52, pp. 169–180, 2005.

[47] T. C. Du, E. Y. Li, and A.-P. Chang, "Mobile agents in distributed network management," *Communications of the ACM*, vol. 46, no. 7, pp. 127–132, 2003.

[48] G. Berbeglia, J.-F. Cordeau, and G. Laporte, "Dynamic pickup and delivery problems," *European Journal of Operational Research*, vol. 202, no. 1, pp. 8–15, 2010.

[49] M. Schneider-Fontan and M. J. Mataric, "Territorial multi-robot task division," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 5, pp. 815–822, 1998.

[50] A. Howard, M. J. Matarić, and G. S. Sukhatme, "Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem," in *Distributed Autonomous Robotic Systems 5*, pp. 299–308, Springer, 2002.

[51] J. Cortés, S. Martínez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 2, pp. 243–255, 2004.

[52] S. Poduri and G. S. Sukhatme, "Constrained coverage for mobile sensor networks," in *IEEE International Conference on Robotics and Automation*, pp. 165–171, 2004.

[53] M. Schwager, D. Rus, and J.-J. Slotine, "Decentralized, adaptive coverage control for networked robots," *International Journal of Robotics Research*, vol. 28, no. 3, pp. 357–375, 2009.

[54] L. Pimenta, V. Kumar, R. C. Mesquita, and G. Pereira, "Sensing and coverage for a network of heterogeneous robots," in *IEEE Conference on Decision and Control*, pp. 3947–3952, 2008.

[55] J. W. Durham, R. Carli, P. Frasca, and F. Bullo, "Discrete partitioning and coverage control with gossip communication," in *ASME Dynamic Systems and Control Conference*, pp. 225–232, 2009.

[56] S. Yun and D. Rus, "Distributed coverage with mobile robots on a graph: Locational optimization," in *IEEE International Conference on Robotics and Automation*, pp. 634–641, 2012.

[57] M. Zhu and S. Martínez, "Distributed coverage games for energy-aware mobile sensor networks," *SIAM Journal on Control and Optimization*, vol. 51, no. 1, pp. 1–27, 2013.

[58] S. Lloyd, "Least squares quantization in pcm," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, 1982.

[59] J. Cortes, S. Martinez, and F. Bullo, "Spatially-distributed coverage optimization and control with limited-range interactions," *ESAIM: Control, Optimisation and Calculus of Variations*, vol. 11, no. 4, pp. 691–719, 2005.

[60] A. Kwok and S. Martínez, "Deployment algorithms for a power-constrained mobile sensor network," *International Journal of Robust and Nonlinear Control*, vol. 20, no. 7, pp. 745–763, 2010.

[61] G. Arslan, J. Marden, and J. S. Shamma, "Autonomous vehicle-target assignment: a game theoretical formulation," *ASME Journal of Dynamic Systems, Measurement, and Control*, pp. 584–596, 2007.

[62] J. R. Marden and A. Wierman, "Distributed welfare games with applications to sensor coverage," in *IEEE Conference on Decision and Control*, pp. 1708–1713, 2008.

[63] A. Arsie, K. Savla, and E. Frazzoli, "Efficient routing algorithms for multiple vehicles with no explicit communications," *IEEE Transactions on Automatic Control*, vol. 54, no. 10, pp. 2302–2317, 2009.

[64] J. Reese, "Solution methods for the p-median problem: An annotated bibliography," *Networks*, vol. 48, no. 3, pp. 125–142, 2006.

[65] N. Megiddo, E. Zemel, and S. L. Hakimi, "The maximum coverage location problem," *SIAM Journal on Algebraic Discrete Methods*, vol. 4, no. 2, pp. 253–261, 1983.

[66] S. Khuller, A. Moss, and J. S. Naor, "The budgeted maximum coverage problem," *Information Processing Letters*, vol. 70, no. 1, pp. 39–45, 1999.

[67] S. H. Owen and M. S. Daskin, "Strategic facility location: A review," *European Journal of Operational Research*, vol. 111, no. 3, pp. 423–447, 1998.

[68] A. Caprara, P. Toth, and M. Fischetti, "Algorithms for the set covering problem," *Annals of Operations Research*, vol. 98, no. 1-4, pp. 353–371, 2000.

[69] Z. Abrams, A. Goel, and S. Plotkin, "Set k-cover algorithms for energy efficient monitoring in wireless sensor networks," in *International Symposium on Information Processing in Sensor Networks*, pp. 424–432, 2004.

[70] M. Charikar, S. Guha, É. Tardos, and D. B. Shmoys, "A constant-factor approximation algorithm for the k-median problem," *Journal of Computer and System Sciences*, vol. 65, no. 1, pp. 129 – 149, 2002.

[71] C. Blum and A. Roli, "Metaheuristics in combinatorial optimization: Overview and conceptual comparison," *ACM Computing Surveys (CSUR)*, vol. 35, no. 3, pp. 268–308, 2003.

[72] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: optimization by a colony of cooperating agents," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 26, no. 1, pp. 29–41, 1996.

[73] H. Mühlenbein, M. Gorges-Schleuter, and O. Krämer, "Evolution algorithms in combinatorial optimization," *Parallel Computing*, vol. 7, no. 1, pp. 65–85, 1988.

[74] H. R. Lourenço, O. C. Martin, and T. Stützle, "Iterated local search," *International Series in Operations Research and Management Science*, pp. 321–354, 2003.

[75] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simmulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

[76] F. Glover, "Tabu search–part i," *ORSA Journal on Computing*, vol. 1, no. 3, pp. 190–206, 1989.

[77] E. Klavins, R. Ghrist, and D. Lipsky, "A grammatical approach to self-organizing robotic systems," *IEEE Transactions on Automatic Control*, vol. 51, no. 6, pp. 949–962, 2006.

[78] Y. Suhov and M. Kelbert, *Probability and Statistics by Example: Volume 2, Markov Chains: A Primer in Random Processes and Their Applications*, vol. 2. Cambridge University Press, 2008.

[79] R. B. Ash, *Basic probability theory*. Courier Dover Publications, 2012.

[80] G. F. Lawler, *Introduction to stochastic processes*. CRC Press, 1995.

[81] L. Jia, R. Rajaraman, and T. Suel, "An efficient distributed algorithm for constructing small dominating sets," *Distributed Computing*, vol. 15, no. 4, pp. 193–205, 2002.

[82] F. Kuhn and R. Wattenhofer, "Constant-time distributed dominating set approximation," *Distributed Computing*, vol. 17, no. 4, pp. 303–310, 2005.

[83] J. Vermorel and M. Mohri, "Multi-armed bandit algorithms and empirical evaluation," in *European Conference on Machine Learning*, pp. 437–448, 2005.

[84] L. E. Blume, "The statistical mechanics of strategic interaction," *Games and Economic Behavior*, vol. 5, no. 3, pp. 387–424, 1993.

[85] J. R. Marden and J. S. Shamma, "Revisiting log-linear learning: Asynchrony, completeness and payoff-based implementation," *Games and Economic Behavior*, vol. 75, no. 2, pp. 788–808, 2012.

[86] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2508–2530, 2006.

[87] R. Tarjan, "Depth-first search and linear graph algorithms," *SIAM Journal on Computing*, vol. 1, no. 2, pp. 146–160, 1972.

[88] H. P. Young, "The evolution of conventions," *Econometrica: Journal of the Econometric Society*, vol. 61, no. 1, pp. 57–84, 1993.