ABSTRACT

Title of dissertation: CONSENSUS PROBLEMS AND
THE EFFECTS OF GRAPH TOPOLOGY
IN COLLABORATIVE CONTROL

Pedram Hovareshti, Doctor of Philosophy, 2009

Dissertation directed by: Professor John S. Baras
Department of Electrical and Computer Engineering

In this dissertation, several aspects of design for networked systems are addressed. The main focus is on combining approaches from system theory and graph theory to characterize graph topologies that result in efficient decision making and control. In this framework, modelling and design of sparse graphs that are robust to failures and provide high connectivity are considered.

A decentralized approach to path generation in a collaborative system is modelled using potential functions. Taking inspiration from natural swarms, various behaviors of the system such as target following, moving in cohesion and obstacle avoidance are addressed by appropriate encoding of the corresponding costs in the potential function and using gradient descent for minimizing the energy function. Different emergent behaviors emerge as a result of varying the weights attributed with different components of the potential function. Consensus problems are addressed as a unifying theme in many collaborative control problems and their robustness and convergence properties are studied. Implications of the continuous convergence property of consensus problems on their

reachability and robustness are studied. The effects of link and agent faults on consensus problems are also investigated. In particular the concept of invariant nodes has been introduced to model the effect of nodes with different behaviors from regular nodes. A fundamental association is established between the structural properties of a graph and the performance of consensus algorithms running on them. This leads to development of a rigorous evaluation of the topology effects and determination of efficient graph topologies.

It is well known that graphs with large diameter are not efficient as far as the speed of convergence of distributed algorithms is concerned. A challenging problem is to determine a minimum number of long range links (shortcuts), which guarantees a level of enhanced performance. This problem is investigated here in a stochastic framework. Specifically, the small world model of Watts and Strogatz is studied and it is shown that adding a few long range edges to certain graph topologies can significantly increase both the rate of convergence for consensus algorithms and the number of spanning trees in the graph. The simulations are supported by analytical stochastic methods inspired from perturbations of Markov chains. This approach is further extended to a probabilistic framework for understanding and quantifying the small world effect on consensus convergence rates: Time varying topologies, in which each agent nominally communicates according to a predefined topology, and switching with non-neighboring agents occur with small probability is studied. A probabilistic framework is provided along with fundamental bounds on the convergence speed of consensus problems with probabilistic switching. The results are also extended to the design of robust topologies for distributed algorithms.

The design of a semi-distributed two-level hierarchical network is also studied,

leading to improvement in the performance of distributed algorithms. The scheme is based on the concept of social degree and local leader selection and the use of consensus-type algorithms for locally determining topology information. Future suggestions include adjusting our algorithm towards a fully distributed implementation.

Another important aspect of performance in collaborative systems is for the agents to send and receive information in a manner that minimizes process costs, such as estimation error and the cost of control. An instance of this problem is addressed by considering a collaborative sensor scheduling problem. It is shown that in finding the optimal joint estimates, the general tree-search solution can be efficiently solved by devising a method that utilizes the limited processing capabilities of agents to significantly decrease the number of search hypotheses.

CONSENSUS PROBLEMS AND THE EFFECTS OF GRAPH TOPOLOGY IN COLLABORATIVE CONTROL

by

Pedram Hovareshti

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2009

Advisory Committee:
Professor John S. Baras, Chair/Advisor
Professor William S. Levine
Professor Nuno C. Martins
Professor Eyad Abed
Professor David Lovell

Dedication

To Banafsheh, Mitra, Maman and Baba

Acknowledgments

It has been a long journey. I am forever indebted to my advisor Professor John S. Baras for his invaluable mentorship, generous support, and enormous motivating drive. His vast vision, deep knowledge, and endless enthusiasm and persistence in solving challenging problems are exemplar and have been a constant source of motivation for me. I am sincerely grateful to him for his understanding and endless support in helping me with dealing with an unexpected situation which was beyond anybody's control. Words fail me when I try to describe my depth of gratitude towards him.

I am grateful to Professors William Levine, Eyad Abed, and Nuno Martins for all I have learnt from them both inside and outside of the classroom. Their instructive suggestions as my dissertation committee members and the time they spent on it have been invaluable to me. I owe much to Professor Abed also through many seminars and workshops he organized at ISR. These occasions gave me the opportunity to learn about many research areas. I also would like to thank Professor David Lovell for agreeing to serve on my dissertation committee.

This work would not have been possible without the help and guidance of Professors Xiaobo Tan and Vijay Gupta during their Post-doctoral stint at UMCP. I am indebted to Professor Tan for his contributions in the initial stage of my research which led to the development of Chapter 2 of this dissertation. I am grateful to Professor Gupta for his help and inspiration with the material of Chapters 4 and especially 6. Special thanks goes to Professor Andre L. Tits who helped and supported me during the times when few would take the responsibility.

I would like to thank my good friends, especially Drs. Maben Rabi and Punyaslok Purkayastha for all the fruitful discussions and support. Many thanks are due for Arash, Azadeh, Bijan, George P., George T., Houman, Nazanin, Negin, Tao, and many more. Your friendship and support has meant a lot to me. I would also like to thank the administrative staff of ISR and ECE for their neverending assistance. I have special thanks for Kim Edwards, Althia Kirlew and Elisabet El Khodary.

And there is, of course, my family that I can hardly find words to express my love and gratitude for them. My lovely wife, Banafsheh, has endured a lot and has shown the brightness to me so patiently at times when all I could see was darkness. She has been my rock throughout this journey. My Mom and Dad have been my first teachers and have continued to support me in every possible way lovingly. My sister has never waivered in her endless support and encouragement of me. Many thanks go to Dayee Parviz and Shirin for all their help and encouragement.

I acknowledge the financial support of my research by the Army Research Office under ODDR&E MURI01 Program Grant NO. DAAD19-01-10465 to the Center for Communicating Networked Control Systems (through Boston University) and Award No DAAD190110494, by the Army Research Laboratory under the Collaborative Technology Alliance program cooperative agreement DAAD19-01-2-001, and by National Aeronautics and Space Administration cooperative agreements NCC8-235 and NAG3-2644.

Table of Contents

List of Figures

Chapter 1

Introduction

1.1 Networked systems and decentralized control

The study of networked systems has gained lots of interest in the recent years. Many applications from different disciplines have emerged, which have a unifying theme in which a group of agents achieve certain objectives via interaction at local levels. The disciplines range from wireless communication to biology and social studies. In most of these applications, existence of a central control unit which coordinates the agents actions is simply not possible. Therefore, the objective has to be achieved through local interactions in a decentralized manner.

In order to understand how these systems function and to be able to design efficient control algorithms to enhance their performance, control and system theorists have recently considered an approach which combines tools from graph theory with traditional system theory [79, 20, 86, 14, 57, 110]. This approach has its roots in the modelling and control of formations and swarms of moving agents and has lead to an active research area, which utilizes consensus algorithms to address many problems of this type. Consensus problems are important on their own right, because of their direct and indirect effect on the study of many collaborative control and computing problems. In addition, they are important since many problems of interest, such as belief propagation and even nonlinear schemes require message passing, for which many properties of diffusion type algorithms

and consensus are inherited.

Our work fits in this framework. Distributed decision making in networked systems relies critically on the timely availability of admissible data. Our main objective is to characterize graph topologies which result in efficient decision making and control. In this avenue, we address the limitations on performance measures based on the graph topology and switching probabilities. We also consider scheduling and switching in the context of performance enhancement in sensor networks. In the sequel, we briefly introduce the main theme of the dissertation.

1.2 Collaborative control of autonomous vehicles

As a key example, in the control community the collaborative control of autonomous unmanned vehicles (AUVs) has sparked lots of challenges due to the vast host of applications they are able to serve. An important source of such applications is exploring uncertain areas. For example, in marine applications the goal is usually to explore undersea in search of resources. In battlefield applications the goal is to move in hazardous terrains to capture unknown areas, while avoiding hostile moving threats and stationary threats and obstacles such as mine fields and protected areas. Recent technological advancements have enabled the use of swarms of small autonomous robots for data gathering and other missions in adversarial environments.

While each application has its own environment-dependent specifications, limitations and prospects, there are a few common aspects which justify the development of a framework for studying the coordination and collaboration of autonomous vehicles as

well as other networked systems. Agents usually face similar limitations. Communication is constrained due to limited available transmission power. Other practical restrictions include limitations in processing power and memory. The performance of autonomous agents depends heavily on their ability to extract, share and convey mission specific information efficiently, as well as their ability to make effective decisions based on the information available through self discovery and neighbor communications. Therefore any successful coordination and control scheme should satisfy three main specifications.

1. Distributed: Using centralized schemes for control is too demanding. The restrictions on computational and communication resources require schemes with the least amount of centralized control. Each agent should use its resources to design control laws which are in line with the system's decentralized information patterns.

2. Fast converging: The decentralized schemes should not only be *correct* but also converge to the desired output as fast as possible. Therefore, we should be able to design coordination topologies which result in effective information dissemination. Meanwhile, by using efficient communication schemes we should minimize the necessity of redundant communication.

3. Robust: The schemes should also be reliable in the sense that they should be robust to failures. These include both agent failures as well as failures in communication and data transmission. Due to uncertain and adversarial nature of battlefield scenarios, communication and agent losses happen frequently and the designed algorithms should be able to perform well in the presence of failures

1.3 Inspirations from nature

Nature is full of representations of emergent phenomena which result from interactions between single "agents" without any central control. Biological systems such as social insects, embryos and bacteria are manifestations of agents which only use local information and reach self organization.

The coordinated behavior of ant colonies emerges as a result of interactions among individual ants with local information access and also their interaction with the environment (Stigmergy) [41, 32]. Ant colonies consist of ants with different jobs or roles in the colony. The queen's job is merely to lay eggs and she does not have any authority. The rest of ants are workers and their job is to forage and gather food, to clean inside the nest and to patrol and find food sources. The organization of the worker ants into groups and the task allocation is done in a decentralized manner. The number of ants which participate in a certain activity is determined from the interaction of the worker ants who sense the chemical contents of other ants, which represent the task they are involved in. Other social insects such as locusts and caterpillars also interact with their 'neighbors' in their groups and their behavior is affected heavily by the way such interactions take place.

Animal aggregation, coordinated motion and swarm formation is not specific to social insects and can be seen in animals of large sizes, such as whales. Flocks of birds, schools of fish and herds of mammals are examples of other animal aggregations. It is widely accepted that synchronization and coordination among animals influences the efficiency of their collaborative actions and therefore is evolutionary beneficial in the sense that increases positive attributes such as survival chance and reproduction success. With

the different kinds of swarming activities in nature it is difficult to build generalizations; however, the swarms in nature usually share the following specifications: 1) Large scale coordination in motion results from local interactions; the scale of interaction is much smaller than the size of the swarm 2) No central entity controls the individuals 3) The size and the shape of the swarm are emergent properties which are a function of the resources and environment. 4)The aggregation is usually stable in normal conditions even when individuals are excluded for some reason. [88, 19]

As mentioned above,'Swarming by nature' has its engineering counterpart, 'swarming by design'. Swarming by design is a fast emerging vast field of research that studies the emergent behavior in large scale networks of vehicles or other agents. There are some similarities and differences between these two big classes of swarms. Therefore, although there are many avenues to get inspiration from natural swarms for solving engineering problems, some limitations exist. Natural systems usually consist of multiple self-interested agents which collaborate out of necessity with close neighbors, and leave traces in environment in order to gain an accepted level of utility (fitness with regard to some measure). Similarly, in engineered systems, we usually have self interested agents with necessary sensing and actuation apparatus and want them to have a desired level of performance. The major differences are that biology usually concerns a large number of agents and also requires a long time span to converge to interesting results, whereas in engineering there are operational limitations on the number of agents and also we are interested in results in a much shorter time span. Despite the limitations, there is much to learn from nature. Many efforts have been done to study the emergent phenomena in nature. The dynamics of groups of aggregated animals have been the subject of many

recent research efforts. (See the reference [19] and the references therein.)

Inspired by the emergent behaviors in nature, control methods that yield desired collective behaviors based on local interactions have received great interest [90, 71, 89, 57, 109]. Artificial potentials or digital pheromones are often involved in such methods for multi-vehicle control, see e.g., [71, 85, 5, 89] and the references therein. The potential function method has been used in various robotic applications [97], where the force or other input (e.g., the velocity) is derived from some potential function that encodes relevant information about the environment and the mission. The potential function method can be also used to model the interaction between the agents, e.g. the equations that govern the convergence of the headings of a group of moving objects, can be derived from optimization using energy functions. This is the basis of many recent studies in the modelling of flocking [83, 26, 110, 109] and leads to the study of *consensus* problems.

1.4 Consensus problems

Consensus problems appear in many applications in collaborative control in which autonomous agents need to coordinate. Consensus algorithms are examples of emergence of global collective behavior using local interactions and arise naturally in networked systems. As pointed out by [95], the notion of a "coordination variable" is common in many instances of collaborative control problems. This corresponds to an amount of information that must be jointly shared by agents to achieve a common goal. For example, in leader-follower schemes in formation control, the state of the leader agent is the coordination variable, since once all the vehicles know this information the course of their action

is completely decided. Some other examples of coordination variables are the common heading of a group of aerial vehicles [9], the estimated value of a quantity measured by a sensor network [123], the common time to finish a task, etc. In all of the applications, it is desired to design consensus protocols that result in meaningful coordination variables for a group of agents in a timely manner and are robust to communication and agent failures.

The problem of reaching agreement by sharing information has been subject of research in statistics, game theory, and distributed computation as well as control theory. DeGroot [43] studied statistical consensus in the context of a group of experts who want to come up with a decision. In a celebrated paper, Aumann [3] provided a definition for common knowledge and proved that for agents with common priors, if their posteriors are common knowledge, then they must agree. In other words they cannot agree to disagree. Aumann's paper initiated a large body of research, which lead to some key papers on distributed estimation and detection in the control community. [16, 111, 115, 114]. Consensus protocols have also been studied for load balancing and distributed computing [27, 11].

Recently, two major classes of applications have caused lots of attention to consensus problems in collaborative control. The first class, as mentioned before comes from formation control, flocking, rendezvous in space and navigation applications. The second class of applications, come from sensor network applications and include distributed sensor fusion and gossip algorithms. We briefly mention the principal works which were crucial in forming the framework for the study of consensus problems in collaborative control.

A. **Vicsek's model for leaderless coordination [117] [57] [86]**

Based on a computer animation scheme proposed by Reynolds [96], Vicsek et al. [117]proposed a simple discrete time model of n autonomous agents all moving in the plane with the same speed but with different headings. Each agent's heading is updated using a local rule based on the average of its own heading plus the headings of its neighbors. Jadbabaie et al. [57] studied a simplified version of Vicsek's model. In the most general directed neighborhood version, we can write the evolution of the agents' heading as:

$$\theta_i(t+1) =< \theta_i(t) >= \frac{1}{n_i(t)}[\sum_{j \in N_i(t)} \theta_j(t)] \tag{1.1}$$

in which agent i's neighbors at time t are those agents including itself which are inside a circle of pre-specified radius r_i centered at agent i's current position. Let $\mathbb{G}$ be the set of directed graphs on n vertices with at most one edge from any vertex to another and exactly one self edge from any vertex to itself. Let P be a suitably defined set that indexes set $\mathbb{G}$. For each $p \in P$ define a "flocking" matrix

$$F_p = D_p^{-1} A_p', \tag{1.2}$$

where A_p is the adjacency matrix and D_p is a diagonal matrix with the degrees of nodes on the diagonal. [1] Now one can rewrite the set of equations (1.1) as

$$\theta(t+1) = F_{\sigma(t)}\theta(t) \tag{1.3}$$

This way the simplified Vicsek's model is represented as a switched linear system whose switching signal takes values in a set of indices that parameterize the set of underlying graphs.

[1]The graph theory terminology is included in the Appendix

Olfati-Saber and Murray studied a continuous time version of the model and its variants in the presence of time delays and link weights [86]. Their basic set up is a network of integrators where the dynamics of each agent change as:

$$\dot{x_i} = \sum_{j \in N\{i\}} (x_j - x_i) \tag{1.4}$$

They showed that in switching topologies the system can be represented as:

$$\dot{x} = -L_{\sigma(t)} x \tag{1.5}$$

where D and A are defined similar to D_p and A_p and $L = D - A$ is the Laplacian of the underlying graph. They showed that the speed of convergence of the system is governed by the second smallest eigenvalue of the Laplacian which is also known as the Fiedler eigenvalue in the literature.

B. **Distributed sensor fusion and gossip algorithms [123] [17]**

Consider the estimation of an unknown parameter $\theta \in R$ using a network of n distributed sensors. Each sensor makes a noisy measurement

$$y_i = \theta + v_i \qquad i = 1, ..., n \tag{1.6}$$

where the noises are i.i.d. Gaussian: $v_i \sim N(0, \sigma^2)$. The maximum likelihood estimate of θ given the measurements is given by

$$\hat{\theta}_{ML} = 1/n \mathbf{1}^T y \tag{1.7}$$

where $\mathbf{1}$ denotes the vector with all components one. In other words the ML-estimate is the average of the measurements y_i at all sensors. The associated mean

square error is σ^2/n. A distributed linear iterative method can be used to compute the average. At $t = 0$ each node initializes its state to $x_i(0) = y_i$. At each following step each node updates its state with a linear combination of its own state and the states of its neighbors.

$$x_i(t+1) = f_{ii}(t)x_i(t) + \sum_{j \in N(i)} f_{ij}(t)x_j(t) \tag{1.8}$$

Now consider a fixed topology. Then we can write the above equation as:

$$x(t+1) = Fx(t) \tag{1.9}$$

It can be shown that in the case that F is a doubly stochastic matrix, iteration(1.1) will result in

$$\lim_{N \to \infty} F^N = \frac{1}{n}\mathbf{1}\mathbf{1}^T \tag{1.10}$$

So as $N \to \infty$ the value of all nodes will converge to the average of their initial value which is the same as the ML- estimate. The scheme can be generalized to contain more realistic scenarios.

There are many interesting problems related to consensus schemes and their convergence. A large body of research has addressed the convergence of consensus schemes under different connectivity and information flow conditions. [13, 57, 78, 79, 95, 37]. The reference [37] provides an interesting survey of the results. In order to be able to design and compose efficient algorithms, an important aspect of research is to explore the convergence properties of consensus algorithms, such as the rate of convergence and robustness to malfunctions. A by-product of such study is that it will reveal the limitations

of consensus schemes for certain applications, which will cause the designers to bring forth plans to overcome the limitations.

Furthermore, the close connections between consensus schemes and Markov chains instigates the use of matrix theory and properties of Markov chains to investigate the structural properties of the connectivity networks. This is a useful tool in the study of complex networks. It is noteworthy to observe that some structural characteristics of complex networks such as the small world phenomenon can be expressed as properties of consensus algorithms running on them. Moreover, in the presence of 'hubs' or 'social leaders' consensus type algorithms can be used to provide each agent with some understanding of the global structure of the graph. The nodes can use this information to divide the original graph to smaller subgraphs on which decentralized algorithms converge much faster, while the connectivity of the whole graphs will be maintained on the hub level.

1.5 Robust communication topologies for distributed algorithms

Different performance measures can be defined for distributed algorithms, the most important of which are the speed of convergence, robustness to link/agent failure, and energy/communication efficiency. These performance measures cannot be achieved all at once and there is a trade off between the level at which the various measures can be fulfilled. All of these measures depend substantially on the structure of the network that the algorithm is running on as well as the dynamics of the system. The problem of identification and design of networks with robust structure in the presence of connectivity constraints is therefore very important.

In a cooperative system of autonomous agents, it is appropriate that the agents form a network which is robust to link losses. The network formation is usually modelled sequentially. Local schemes −which use locally available information− have attracted much interest. Spanos and Murray [103] consider a localized notion of connectedness and study its relationship to the global connectivity of a network of vehicles. Zavlanos and Papas [124] address the problem of controlling the motion of a network of agents while preserving $k-$hop connectivity. Das and Mesbahi [28] have studied the problem of transmit power optimization with $k-$node connectivity constraint in a wireless framework using semidefinite programming.

Meanwhile, it is also important to address the effect of adding links on the global measures of network −robustness− even in the cases where local computation of such global measures seems infeasible. The reason is twofold. First, it provides upper bounds on the improvement based on local schemes, determines fundamental limitations of the design, and provides benchmarks for comparison of local measures. Second, since network formation is a gradual process it is plausible that the nodes initially have some information about the network structure and use this information in the process of edge augmentation. Also, there exist methods which provide nodes with information on the global topology based on local message passing algorithms. A valid question is then given the present structure of the network and constraints on link establishment how should a node choose which link to establish in order to maximize a global measure of network robustness? To answer this question, it is important to notice that autonomous agents are critically influenced by their understanding of the network topology. Therefore, their behavior and performance are functions of their initial knowledge or estimate of the group's

topology.

Ghosh and Boyd [38] consider the optimization of the second smallest eigenvalue of the graph Laplacian as a measure of well-connectedness of the graph. They relaxed the combinatorial problem to a convex problem, used semidefinite programming to solve it, and provided a heuristic for large scale graphs. The Fiedler eigenvalue is a global measure of how fast local diffusion-type algorithms converge on a graph. It also provides a lower bound on the graph's edge and node connectivity. The number of spanning trees of a graph is a more general measure of graph connectivity. This number depends on the value of all the eigenvalues of the Laplacian matrix rather than only the Fiedler eigenvalue and therefore is a more informative measure. As an example, in symmetric graphs such as rings the Fiedler eigenvalue has multiplicity of two or more. As a result its value, does not change with augmentation of an edge. However, adding an edge definitely changes the structure of a graph and its properties. On the other hand, this change is captured in the number of spanning trees of the graph.

1.6 Scheduling for optimal sensor performance

Distributed sensing is another example of a collaborative task. A collaborative sensor network consists of sensors cooperating to estimate a process by exchanging measurements and other data. Since the goal of the network is to achieve better estimation accuracy, a trade-off exists between the complexity of the system and its performance. The communication constraints of the system limit the number of simultaneous sensor transmissions and therefore it is necessary that the sensors send their data based on some

schedule. Echo-based sensors with interference and radar based tracking applications in which different types of measurements are made by transmitting suitable waveforms are cases that represent this need. Shared communication resources that constrain the usage of many sensors at the same time may also be considered as a reason for sensor scheduling.

In networked control systems, sensors are usually equipped to communicate over wireless channels or communication networks. Therefore, it is reasonable to assume that they possess some storage and processing capabilities. If the sensors can execute simple recursive algorithms to process the information being collected, significant improvement in estimation or control performance can be expected. This possibility has been developed in the general framework and methods of [48], where the estimator has complete past histories of measurements, and where even simultaneous measurements by several sensors in each time step are allowed. A challenge in this respect is to come up with optimal sensor schedules which requires pruning a decision tree efficiently to avoid memory and computation complexities.

1.7 Contributions

In this dissertation certain aspects of collaborative control have been studied. First, it is shown that a decentralized approach to path generation can be used to observe different behaviors among unmanned autonomous agents using potential functions. If the agents' mission is to cover a target area while moving in cohesion and avoiding obstacles and threats, it is shown that these goals can be satisfied via correct encoding into a poten-

tial function and the use of a gradient descent method for minimizing the energy function. Results show that the decentralized approach leads to interesting emergent behaviors, and the behaviors can be varied by adjusting the weighting coefficients of different potential function terms.

In particular, consensus-type schemes can be viewed as gradient descent algorithms on graphs, in which a 'Laplacian' energy function is minimized by local interaction between the agents. Our main focus in this dissertation has been on the properties of convergence of these schemes, which make them desirable for many applications. We have considered the performance measures of these schemes to be the speed of convergence and the deviation of the schemes' results as a response to changes in agents behavior. These correspond to two main issues in the design of control algorithms: convergence rate and robustness.

The convergence speed of consensus-type algorithms is determined by two essential factors: graph topology and the weights that each agent applies to the inputs from other agents. Graphs which are 'better connected' result in higher convergence rates. Since agents usually have energy or other constraints, the number of agents with which they can communicate and the 'long-range' interconnections that they can maintain is limited. Thus it is important to investigate convergence over 'efficient' (from a communications resources perspective) graph topologies.

1.7.1 Consensus problems on small world graphs

The models of complex networks in engineering, computer science, biology, social sciences and other branches of science range from regular networks to completely random networks. If we take regular networks and random networks as idealizations which make analysis tractable in many applications, many real world networks lie somewhere in between complete order and complete randomness. Watts and Strogatz [118] introduced and studied a simple tunable model that can explain behavior of many real world networks for which some notion of efficiency is crucial. Starting from a regular lattice, their small world model replaces the original edges by random edges with some probability. They found that even with small probability of rewiring, the network will transform to a "small world" in which short paths can be found between nodes and the network remains clustered. Watts and Strogatz conjectured that dynamic systems coupled in this way would enjoy enhanced signal propagation and global coordination compared to lattices of the same size. In the context of continuous time consensus protocol, this was shown to be true by the simulation studies of Olfati-Saber [82]. Our simulation results show that this property holds also for discrete time consensus algorithms.

Our main contribution in this part is to show the truth of this conjecture in the context of consensus problems. We develop a method for investigating the effects of small world topologies by building on the probabilistic models of Higham [55], that established an equivalent representation of small world topologies as rare transitions among non-neighboring states in the Markov chain associated with a graph. In this model and associated method, weak interaction is assumed between the non-neighbor nodes that can

be considered as a mean field approximation for the small world graphs. We show that for very small perturbation of the values of the weights, there is a marked increase in the rate of convergence when starting from lattice topologies. By performing a quantitative analysis of the eigenvalues of the resulting matrices and by employing an appropriate parametrization of these small positive weights, a complete characterization is given as to when small world phenomena (manifested by convergence speed-up) will occur.

This interpretation prompts a probabilistic viewpoint towards understanding and quantifying small world effects on consensus convergence rates: we utilize time-varying topologies, in which every node nominally communicates according to a pre-defined topology, corresponding to an original base graph from which a small world network is obtained. Apart from these fixed connections, each node communicates with a small probability with the non-adjacent nodes at every time step. Since the small world model is obtained by stochastically adding or rewiring a few edges to a base graph, we anticipate that adding a small number of long distance edges is analogous to choosing graphs with low probability shortcuts. We determine lower and upper bounds for the convergence rate of consensus algorithms on the resulting graphs. The developed method is also used for investigating the convergence rate of consensus algorithms in other occasions that probabilistic switching may occur, e.g. how will probabilistic link losses affect the convergence rate of consensus algorithms running on a given topology.

1.7.2 Robustness, invariant nodes, and hierarchical network design scheme

When a consensus method is considered for a certain application, it is important to investigate what happens if some of the links or agents fail. The robustness of the decentralized scheme to links and agent failures has been investigated. In particular we have considered the case with exogenous agents who do not follow the regular dynamics of the consensus scheme. We call such exogenous agents invariant nodes, since their dynamics is not affected by the scheme. Based on the application these nodes can be thought of as local leaders, adversaries or intruders trying to break the group's consensus or simply nodes of more importance than regular nodes. Examples include a flock of birds with local leaders trying to move the flock toward different directions, a swarm of insects being navigated by sensing different food source gradients, sources of authentication in a trust establishing scheme or even intruders or malicious agents in such systems.

We have analyzed the stability and convergence properties of networks of dynamic agents in the presence of invariant nodes. We show that determination of the steady state values of regular nodes in the presence of invariant nodes results in solving a discrete Dirichlet problem with boundary values given by the invariant nodes. Its implication is that consensus algorithms are not robust to the failures in which a group of nodes assert their values without considering the group protocol, which includes the single node failures. However, such failures are detectable in cases where the participating nodes are well-connected in the graph.

We have used the properties of invariant nodes within the framework of consensus problems to propose a hierarchical network design algorithm aimed at running decen-

tralized algorithms in a fast and efficient manner. The idea is based on breaking a large graph to small subgraphs centered around more influential nodes and make the "inter-community" communication take place through these hubs. After hub selection is done in the first stage of the algorithm, the hubs can be considered as invariant nodes and regular nodes utilize a consensus-type algorithm to determines which subgraph they should be affiliated with. The unnecessary links which decelerate the speed of system can then be removed. In a broader view, the small world phenomenon as it happens in social systems can be thought of as a result of a similar procedure: communities shape based on local needs and local connectivity; whereas inter-community connectivity is maintained through fewer yet more reliable means. Extracting efficient community structures in graphs is useful in many applications, e.g. in backbone construction in wireless ad hoc networks [70].

1.7.3 Robust communication topologies for distributed algorithms

We address the problem of network formation with robustness and connectivity constraints. Considering the number of spanning trees of a graph as a performance measure, we show that the general combinatorial problem can be relaxed to a convex optimization problem. We also show that two issues of symmetrizing the graph and reducing graph's effective resistance distance appear in the problem of maximizing the number spanning trees and the optimal graph can be considered as a result of the interaction of these two factors. This is reminiscent of the logic behind the formation of small world graphs which is a trade-off between increasing clustering and decreasing distance. We explicitly solve

the problem for the special case of adding a shortcut to a ring and analyze the small world effect in the context of abrupt increase in the number of spanning trees as a result of adding shortcuts to a ring in the Watts-Strogatz framework. We use the analysis to provide insights for derivation of heuristics for the general case of optimal edge attachment.

1.7.4 Scheduling with smart sensors

The problem of collaborative sensor scheduling using sensors with processing abilities has also been considered. Based on the methods developed in [44] that proposes a framework for cooperative sensor fusion while utilizing simple processing and memory capabilities of sensors, we have considered the problem of finding the optimal joint estimates. We show that the problem reduces to considering the problem of finding the optimal sensor schedule. While the general solution remains a tree-search, we devise an efficient method to prune the tree, so that the number of paths to be searched is significantly decreased. We also prove a periodicity result in the optimal sensor schedule when there is a single sensor involved, but has constraints on the number of transmissions due to energy or other constraints.

1.8 Outline

The organization of the dissertation is as follows: Chapter 2 considers a general framework of potential functions for problems in collaborative control. Different behaviors in a path planning problem are discussed, a framework for the study of consensus problems is provided and their robustness limitations are discussed. Chapter 3 provides

the background for studying the convergence of consensus problems and design of efficient topologies. The problems of convergence of consensus algorithms on small world networks, probabilistic switching, and design of robust network topologies are addressed in Chapter 4. Chapter 5 considers an efficient hierarchical network design scheme. In chapter 6 the problem of sensor scheduling with smart sensors is studied. Conclusions and future work are included in Chapter 7.

Chapter 2

Potential functions and collaborative control of autonomous agents

In this chapter the basic framework for studying the problems of collaborative control is introduced and several challenges inherent to these type of problems are studied. We follow the artificial potential method, which encodes the local information-based goals of the agents as a potential function. The evolution of the system is then modelled using the gradient descent method. We also consider consensus problems as minimizing potential functions defined on graphs.

A decentralized approach to path generation for a group of autonomous vehicles in a battlefield scenario is first explored using the potential function method. The mission is to maneuver the vehicles to cover a target area while avoiding obstacles and threats. At every time instant each vehicle evaluates its potential function profile and decides its velocity using the gradient descent method. The potential function consists of several terms reflecting the objective and the constraints. It is constructed so that only information about neighboring vehicles, local information about dynamic threats, and some static information (about stationary threats, targets) are involved.

The existence of local minima produces some challenges for the path planning problems. In particular, the behavior of a vehicle experiencing both attraction from the target and repulsion from the obstacles may cause it to fall into a local minimum of the potential. This is studied through the vector field analysis. Some emergent behaviors of the

vehicles are studied and their qualitative behaviors are discussed. In particular, simulation results show that the decentralized approach leads to interesting emergent behaviors, and the behaviors can be varied by adjusting the weighting coefficients of different potential function terms.

Based on the inter-agent artificial potential formulation, we consider the consensus problems as gradient descent of a harmonic potential profile on graphs. The basic framework and notations for continuous consensus problems and their robustness properties will also be addressed at the dissertation. The chapter is mainly based on [7].

This chapter is organized as follows. In section 2.1 the problem setup is described and the potential functions constructed. Analysis of vehicle behaviors is performed in Section 2.2. Simulation results are reported in Section 2.3. The flocking models and their connection to consensus problems are addressed in Section 2.4. Finally, the concept of invariant nodes is introduced and their application to robustness of consensus problems is studied in Section 2.5.

2.1 Potential Functions

We study the kinematic planning problem for N vehicles moving on a (two dimensional) plane. Extension to three dimensional space is straightforward, although the analysis will be more complicated. The task for the vehicles is to move toward and then occupy a connected target area $A \subset \mathbb{R}^2$. They should avoid crashing into obstacles that are distributed in the battlefield. There are also threats, both stationary ones and moving ones, that endanger the vehicles if they are close. It is assumed that each vehicle has

the knowledge of locations of stationary threats. A vehicle detects a moving threat if the threat is within the distance R_d, and is destroyed by the threat if the distance between them is less than R_e ($< R_d$). The vehicles can communicate with each other and exchange information about their positions if they are within the neighboring distance R_c. There is a desired inter-vehicle distance r_0 (less than R_c) for several reasons: staying too close leads to small area of coverage, good chance of collision, and easy targeting by enemy fire, while staying too far apart leads to loss of communication and coordination.

We order the vehicles and identify each vehicle with its index. Each vehicle is treated as a point. Denote the position of vehicle i at time t as $p_i(t) = (x_i(t), y_i(t))$. Let $\mathcal{V}(t)$ be the set of vehicles functioning at time t, and $\mathcal{N}_i(t)$ be the set of the neighbors of vehicle i defined by $\mathcal{N}_i(t) \triangleq \{j \in \mathcal{V}(t) : j \neq i, \|p_i(t) - p_j(t)\| \leq R_c\}$. We also use the notation $j \sim i$, if $j \in \mathcal{N}_i$. From the previous discussions, there are multiple objectives/constraints when a vehicle makes the moving decision. To accommodate this a potential function is constructed for each vehicle that consists of several terms, each term reflecting a goal or a constraint. To be specific, the potential function $J_{i,t}(p_i)$ for vehicle i at t is expressed as

$$J_{i,t}(p_i) = \lambda_g J^g(p_i(t)) + \lambda_n J^n_{i,t}(p_i(t)) + \lambda_o J^o(p_i(t)) + \lambda_s J^s(p_i(t)) + \lambda_m J^m_t(p_i(t)), \quad (2.1)$$

where $J^g, J^n_{i,t}, J^o, J^s, J^m_t$ are the components of the potential function relating to the target, neighboring vehicles, obstacles, stationary threats, and moving threats, respectively, and $\lambda_g, \lambda_n, \lambda_o, \lambda_s, \lambda_m \geq 0$ are the corresponding weighting coefficients. The velocity $\dot{p}_i$ is then given by

$$\dot{p}_i(t) = -\frac{\partial J_{i,t}(p_i)}{\partial p_i}. \quad (2.2)$$

We now describe in detail the components of $J_{i,t}$:

(1) The target potential J^g. $J^g(p_i) = f_g(\rho(p_i, A))$, where $\rho(p_i, A) = \inf_{a \in A} \|p_i - a\|$ (the distance from p_i to the target area A), $f_g(\cdot)$ is a strictly increasing function, and $f_g(0) = 0$. This guarantees that in the absence of other objects, the vehicle will move toward the target. For analysis and simulation in this paper, we choose $f_g(r) = r^2$;

(2) The neighboring potential $J^n_{i,t}$:

$$J^n_{i,t}(p_i) = \sum_{j \in \mathcal{N}_i(t)} f_n(\|p_i - p_j(t)\|),$$

where $f_n : \mathbb{R}^+ \to \mathbb{R}$ is a differentiable function that has the following properties: a) $f_n(r)$ approaches infinity as $r \to 0$ and is strictly decreasing on $[0, r_0]$; b) it is strictly increasing on $[r_0, R_c]$, and $\frac{df_n}{dr} = 0$ on $[R_c, \infty)$. These properties enable two vehicles to maintain optimal mutual distance in the absence of other objects, and to make the transition of dynamics seamless when the neighboring set of a vehicle is changing. An appropriate combination of $\frac{1}{r^2}$, $(r - r_0)^2$, and $-(r - R_c)^2$ is used for f_n in our simulations. An example of such neighboring potential function and its derivative is shown in Figure 2.1, where $r_0 = 2$, $R_c = 8$;

(3) The obstacle potential J^o. An obstacle is a connected, closed set (could be a single point) that a vehicle cannot enter. Assume that there are a finite number of obstacles $\{O_j\}_{j=1}^{N_o}$. Let $J^o(p_i) = \sum_{j=1}^{N_o} f_o(\rho(p_i, O_j))$, where $\rho(p_i, O_j)$ is the distance from p_i to the set O_j, and $f_o(\cdot) : \mathbb{R}^+ \to \mathbb{R}$ is a strictly decreasing function that satisfies $f_o(r) \to \infty$ as $r \to 0$. In this paper f_o is chosen to be $\frac{1}{r^2}$. The information about obstacles can be obtained beforehand, or it can be available in real time through detection;

(4) The potential J^s due to stationary threats. Stationary threats can be modelled similarly

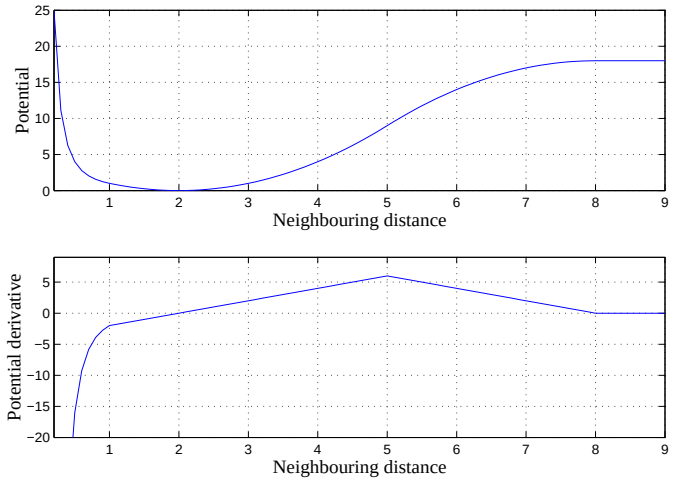


Figure 2.1: A neighboring potential function.

as obstacles, so that vehicles will tend to stay away from them. Anisotropic (direction-dependent) threats can also be included using appropriate potential functions;

(5) The potential J_t^m due to moving threats. A moving threat is modelled as a moving point. Let $\mathcal{M}_i(t)$ be the set of moving threats that are within the detection range of the vehicle i, and q_j be the position of the threat j. Let $J_t^m(p_i) = \sum_{j\in\mathcal{M}_i(t)} f_m(\|p_i - q_j\|)$, where the function $f_m : (R_e, \infty) \to \mathbb{R}$ is differentiable, strictly decreasing on (R_e, R_d), constant on (R_d, ∞), and $f_m(r) \to \infty$ when $r \to R_e$. With this potential function, a vehicle tends to keep at least a distance R_e from moving threats, and its vector field remains continuous when moving threats enter or leave its detection range. A simple example for such $f_m(\cdot)$ is

$$f_m(r) = \begin{cases} \frac{1}{(r-R_e)^2} & \text{if } R_e < r \le \frac{a_2}{2} \\ \frac{16(r-R_d)^2}{a_1^3 a_2} - \frac{8R_e}{a_1^3} & \text{if } \frac{a_2}{2} \le r \le R_d \\ -\frac{8R_e}{a_1^3} & \text{if } r > R_d \end{cases},$$

where $a_1 \stackrel{\triangle}{=} R_d - R_e$, and $a_2 \stackrel{\triangle}{=} R_d + R_e$.

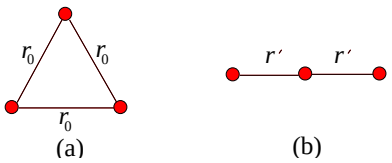


Figure 2.2: Equilibrium configurations for $N = 3$.

2.2 Qualitative Analysis of Agent Behaviors

2.2.1 Equilibrium formations

It is important to study vehicle behaviors under inter-vehicle interactions only. This is especially relevant after the vehicles enter the target area. Here, we assume that the potential function is only composed of J^n.

Proposition 2.2.1. *Let N be the number of vehicles. Then*

(1) the configuration of vehicles converges to some equilibrium;

(2) for $N = 2$, if $\|p_1(0) - p_2(0)\| < R_c$, the vehicles maintain a distance of r_0 in the equilibrium configuration and the equilibrium is globally asymptotically stable;

(3) for $N = 3$, if $\|p_i(0) - p_j(0)\| < R_c$, $1 \leq i, j \leq 3$, the vehicles either form an equilateral triangle (Fig. 2.2(a)), or form a line at the equilibrium. If $\frac{df_n}{dr}$ is strictly increasing on $(0, r_0]$, the collinear configuration is equally spaced with spacing r' (Fig. 2.2(b)), where $\frac{r_0}{2} < r' < r_0$ and $\frac{df_n}{dr}(r') = -\frac{df_n}{dr}(2r')$. Furthermore, if $\frac{df_n}{dr}$ is strictly increasing on $[r_0, 2r_0]$, such r' is unique. The collinear configuration is unstable, while the equilateral configuration is locally asymptotically stable.

Proof. 1. Take the sum J of the neighboring potentials as a candidate Lyapunov func-

tion.

$$J(\{p_i(t)\}) = \frac{1}{2}\sum_{i=1}^{N} J^n(p_i(t), \{p_j(t)\}_{j\neq i}) = \frac{1}{2}\sum_{i=1}^{n}\sum_{j\in\mathcal{N}(i)} f^n(||p_i(t) - p_j(t)||).$$

Since the neighboring potentials are symmetric, taking the derivative results in:

$$\frac{dJ}{dt} = -\sum_{i=1}^{N} ||\frac{\partial J^n}{\partial p_i}(p_i, \{p_j\})||^2.$$

Hence, J is non-increasing with t. Since J is lower bounded, $\frac{dJ}{dt} \to 0$. This implies that $\dot{p}_i \to 0, \forall i$, and therefore the vehicles converge to an equilibrium configuration.

2. For $N = 2$, the potential function is $J(p_1(t), p_2(t)) = f^n(||p_1(t) - p_2(t)||)$. If we denote $r_{12}(t) = ||p_1(t) - p_2(t)||$, then

$$\frac{dJ}{dt} = -2(\frac{df^n}{dr}(r_{12}(t))^2.$$

Therefore $r_{12} \to r_0$ as $t \to \infty$, and this is clearly a stable configuration as long as $||p_1(0) - p_2(0)|| < R_c$.

3. For $N = 3$, the potential function is $J(p_1(t), p_2(t), p_3(t)) = f^n(||p_1(t) - p_2(t)||) + f^n(||p_2(t) - p_3(t)||)f^n + (||p_1(t) - p_3(t)||)$, where $r_{ij}(t) = ||p_i(t) - p_j(t)||$. We can write:

$$\begin{aligned}\frac{dJ}{dt} =& -||\frac{df^n}{dr}(r_{12})\hat{\mathbf{r}}_{12} + \frac{df^n}{dr}(r_{31})\hat{\mathbf{r}}_{13}||^2 - ||\frac{df^n}{dr}(r_{12})\hat{\mathbf{r}}_{21} + \frac{df^n}{dr}(r_{23})\hat{\mathbf{r}}_{23}||^2 \\ & - ||\frac{df^n}{dr}(r_{31})\hat{\mathbf{r}}_{31} + \frac{df^n}{dr}(r_{23})\hat{\mathbf{r}}_{32}||^2, \end{aligned} \tag{2.3}$$

where $\hat{\mathbf{r}}_{ij}$ denotes the vector pointing from p_j to p_i. Equation (2.3) implies that J is strictly decreasing unless one of the two following cases occurs:

- $\frac{df^n}{dr}(r_{12}) = \frac{df^n}{dr}(r_{23}) = \frac{df^n}{dr}(r_{31}) = 0$, which corresponds to the equilateral triangle configuration of Figure 2.2(a).

- The vectors $\hat{\mathbf{r}}_{ij}$ are parallel or antiparallel. Without loss of generality (due to symmetry), we assume that p_2 is between p_1 and p_3, $r_{12} = r_{23} = r'$ for $r' \in (\frac{r_0}{2}, r_0)$ satisfying $\frac{df_n}{dr}(r') = -\frac{df_n}{dr}(2r')$, which corresponds to the collinear configuration of Figure 2.2(b).

□

We note that similar results for the cases $N = 2,\ 3$ also appeared in [5] where second order dynamics and a quadratic potential were considered. For general $N > 3$, one can design the potential function properly so that certain configurations (or *formations*) become equilibria that are *locally* asymptotically stable (also refer to [71] for a discussion on designing stable flocking and schooling motions using "virtual leaders"). For instance, if we design the function f_n with $R_c = \sqrt{3}r_0$, then lattices of equilateral triangles with length r_0 are such equilibria. These equilibria are often desirable: for instance, in the scenario of this chapter, the vehicles in such a condition would provide good area coverage while maintaining optimal inter-vehicle distance. However, due to the existence of multiple locally asymptotically stable equilibria, one cannot guarantee the convergence to a particular desired configuration. Although the ambiguity (of the final formation) can be eliminated using the *structural* potential functions [85], the latter approach requires explicit specification of the communication topology. Such requirement, unfortunately, is not feasible in our scenario, where some of the vehicles might get destroyed during the mission.

Despite the ambiguity problem, extensive simulation appears to support that the final formation is usually well "organized" under purely local interactions. Fig. 2.3 shows

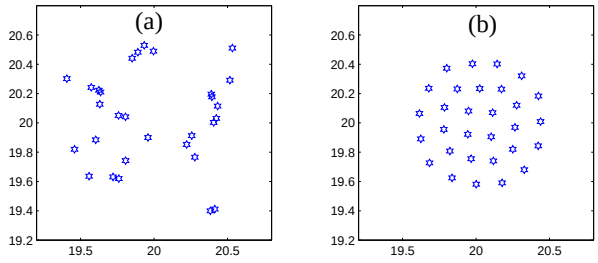


Figure 2.3: Formation of 30 vehicles under local interactions: (a) random initialization; (b) final formation.

the final formation of 30 vehicles starting from a random initialization.

2.2.2 Vector field analysis

Scenario I

In this sub-section we analyze the behavior of a vehicle when it experiences both attraction from a target and repulsion from obstacles. Two scenarios are considered. In the first one (illustrated in Fig. 2.4(a)), the (point) target is located at the origin (0,0), and two (point) obstacles are located symmetrically about the y axis with coordinates $(-a,-b)$ and $(a,-b)$, respectively $(a,b>0)$. The potential function in terms of (x,y) is

$$\lambda(x^2+y^2)+\frac{1}{(x+a)^2+(y+b)^2}+\frac{1}{(x-a)^2+(y+b)^2},$$

and the associated vector field is

$$\begin{cases} \dot{x}(t)=\frac{2(x+a)}{[(x+a)^2+(y+b)^2]^2}+\frac{2(x-a)}{[(x-a)^2+(y+b)^2]^2}-2\lambda x \\ \dot{y}(t)=\frac{2(y+b)}{[(x+a)^2+(y+b)^2]^2}+\frac{2(y+b)}{[(x-a)^2+(y+b)^2]^2}-2\lambda y \end{cases}, \tag{2.4}$$

where the weighting constant for obstacles equals 1. Consider a vehicle initially located on the y axis. We want to know whether it will move toward the target under the vector

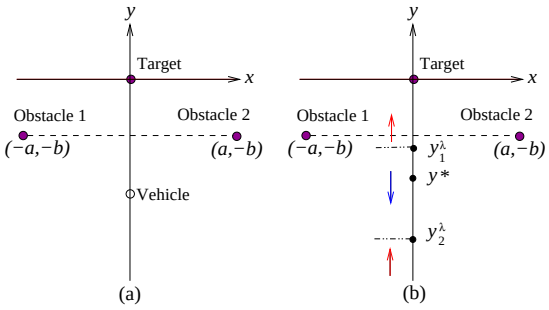


Figure 2.4: (a) The setup of two obstacles and one target; (b) The vector field on the y axis.

field (2.4) when $y < 0$ (the case $y > 0$ is simpler and can be studied similarly). Due to the symmetry, $\dot{x} = 0$, so the real question is whether $\dot{y} > 0$. When $x = 0$,

$$\dot{y} = \frac{4(y+b)}{[(a^2 + (y+b)^2]^2} - 2\lambda y. \tag{2.5}$$

Let $\tilde{y} = y + b$. Obviously, if $\tilde{y} \geq 0$, $\dot{y} > 0$. In the following we study the case $\tilde{y} < 0$.

Proposition 2.2.2. *There is a unique solution* $\tilde{y}^* \in (-\frac{a}{\sqrt{3}}, 0)$ *to*

$$4\tilde{y}^3 - 3b\tilde{y}^2 + a^2 b = 0. \tag{2.6}$$

Also, denoting

$$\lambda^* = \frac{2}{(a^2 + \tilde{y}^{*2})^2}(1 - \frac{4\tilde{y}^{*2}}{a^2 + \tilde{y}^{*2}}), \tag{2.7}$$

and $y^* = \tilde{y}^* - b$*, the following hold:*

(1) If $\lambda > \lambda^*$, $\dot{y} > 0$, $\forall y < -b$*;*

(2) If $\lambda = \lambda^*$, $\dot{y} > 0$ *for* $y \in (-\infty, -b)$ *except at* y^* *where* $\dot{y} = 0$*;*

(3) If $\lambda < \lambda^*$, there exist y_1^λ, y_2^λ dependent on λ, $y_2^\lambda < y^* < y_1^\lambda$, such that

$$\begin{cases} \dot{y} > 0, \text{ if } y \in (-\infty, y_2^\lambda) \\ \dot{y} < 0, \text{ if } y \in (y_2^\lambda, y_1^\lambda) \\ \dot{y} > 0, \text{ if } y \in (y_1^\lambda, -b) \\ \dot{y} = 0, \text{ if } y = y_1^\lambda \text{ or } y_2^\lambda \end{cases},$$

as illustrated in Fig. 2.4(b). Furthermore, as λ decreases from λ^* to 0, y_1^λ increases from y^* to $-b$, and y_2^λ decreases from y^* to $-\infty$.

Proof. Let $h(\tilde{y}) = \frac{4\tilde{y}}{(a^2+\tilde{y}^2)^2}$. Since

$$\frac{dh}{d\tilde{y}} = \frac{4(a^2 - 3\tilde{y}^2)}{(a^2 + \tilde{y}^2)^2}, \tag{2.8}$$

$h(\tilde{y})$ is strictly decreasing on $(-\infty, -\frac{a}{\sqrt{3}})$, and strictly increasing on $(-\frac{a}{\sqrt{3}}, 0)$. From (2.8), $\frac{dh}{d\tilde{y}}$ is also strictly increasing on $(-\frac{a}{\sqrt{3}}, 0)$. Graphical analysis reveals that there exists a unique λ^*, such that the line $l(\tilde{y}) = 2\lambda^*(\tilde{y} - b)$ is tangent to the curve $h(\tilde{y})$ at a unique $\tilde{y}^* \in (-\frac{a}{\sqrt{3}}, 0)$. After algebraic manipulations, one can show that $\tilde{y}^*$ and λ^* satisfy (2.6) and (2.7) respectively. The remaining claims of the proposition follow from the graphical analysis. □

From Proposition 2.2.2, the weight λ determines whether the vehicle can pass the obstacle potential valley and get to the target.

Scenario II

Next we investigate the motion of a vehicle in the presence of one point target (0,0) and one point obstacle $(0, -b)$. Here no constraint on the vehicle position is imposed except that we focus on the region $y < 0$ (the case $y > 0$ is simpler and can be analyzed

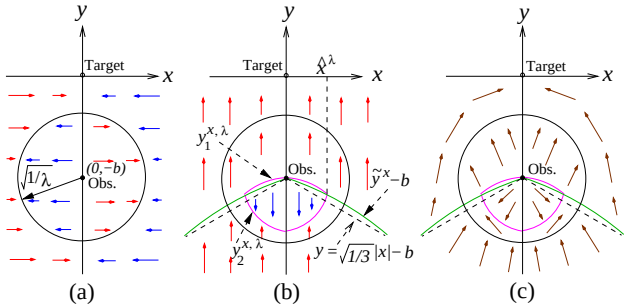


Figure 2.5: Vector field analysis for the case of one obstacle and one target. (a) x-component; (b) y-component; (c) total vector field.

similarly). The vector field is

$$\begin{cases} \dot{x} = \frac{2x}{[x^2+(y+b)^2]^2} - 2\lambda x \\ \dot{y} = \frac{2(y+b)}{[x^2+(y+b)^2]^2} - 2\lambda y \end{cases}. \tag{2.9}$$

We will discuss $\dot{x}$ and $\dot{y}$ separately. Denote by $\mathcal{C}_\lambda$ the circle with radius $\frac{1}{\sqrt{\lambda}}$ centered at $(0,-b)$, $\mathcal{C}_\lambda^-$ the interior of $\mathcal{C}_\lambda$, and $\mathcal{C}_\lambda^+$ the exterior of $\mathcal{C}_\lambda$. Then clearly

$$\begin{cases} \dot{x} > 0, \text{ if } x < 0, (x,y) \in \mathcal{C}_\lambda^+ \text{ or } x > 0, (x,y) \in \mathcal{C}_\lambda^- \\ \dot{x} < 0, \text{ if } x > 0, (x,y) \in \mathcal{C}_\lambda^+ \text{ or } x < 0, (x,y) \in \mathcal{C}_\lambda^- \\ \dot{x} = 0, \text{ if } x = 0 \text{ or } (x,y) \in \mathcal{C}_\lambda \end{cases},$$

as shown in Fig. 2.5(a).

For $\dot{y}$, it's straightforward to verify

$$\dot{y} > 0 \text{ if } (x,y) \in \mathcal{C}_\lambda^+ \text{ or } y \geq -b.$$

However, the analysis is more involved when $y < -b$ and $(x,y) \in \mathcal{C}_\lambda^-$. The proof of the next result shares the spirit of the proof of Proposition 2.2.2:

Proposition 2.2.3. *Let $\tilde{y} = y + b$. For each x, there exists a unique $\tilde{y}^x \in (-\frac{\sqrt{3}}{3}|x|, 0)$ satisfying*

$$4\tilde{y}^3 - 3b\tilde{y}^2 + bx^2 = 0,$$

$\tilde{y}^x = \tilde{y}^{-x}$, and $\tilde{y}^x$ strictly decreases as $|x|$ increases. Let $y^x = \tilde{y}^x - b$. For $\lambda > 0$,there is an $\hat{x}^\lambda > 0$ with $(\hat{x}^\lambda, y^{\hat{x}^\lambda}) \in \mathcal{C}_\lambda^-$, and two continuous functions $y_1^{x,\lambda}$ and $y_2^{x,\lambda}$ of x defined on $[0, \hat{x}^\lambda]$, dependent on λ, that satisfy the following:

(1) $y_1^{x,\lambda}$ decreases as $|x|$ increases, $y_1^{x,\lambda} = y_1^{-x,\lambda}$, $y_1^{x,\lambda} \geq y^x$ where the equality holds only at $x = 0$ and $x = \hat{x}^\lambda$;

(2) $y_2^{x,\lambda}$ increases as $|x|$ increases, $y_2^{x,\lambda} = y_2^{-x,\lambda}$, $y_2^{x,\lambda} \leq y^x$ where the equality holds only at $x = 0$ and $x = \hat{x}^\lambda$.

Denote the region enclosed by the graphs of $y_1^{x,\lambda}$ and $y_2^{x,\lambda}$ as $\mathcal{D}_\lambda$. Then for the case $y < 0$, $\dot{y} \leq 0$ if and only if $(x, y) \in \mathcal{D}_\lambda$, where the equality holds only at the boundary of $\mathcal{D}_\lambda$.

Fig. 2.5(b) illustrates Proposition 2.2.3 and sketches the y-component of the vector field. The total vector field is shown in Fig. 2.5(c). The only point where $\dot{x} = \dot{y} = 0$ is $(0, y_2^{0,\lambda})$. But this is an unstable equilibrium as one can tell from the figure. We can also verify that the linearized system at $(0, y_2^{0,\lambda})$ has a positive eigenvalue. Hence for any $\lambda > 0$, the vehicle will not get blocked by the obstacle potential; but the larger λ, the less "detour" it takes before it moves towards the target.

2.3 Simulation Results

Fig. 2.6 shows the simulation scenario. There are ten vehicles (represented by the pentagons) randomly distributed in the left lower corner at $t = 0$. Two circular obstacles (with radii 3 and 5, respectively) sit between the vehicles and the target (also circular, with radius 1.5). Eight moving threats (represented by the crosses), uniformly distributed around the target, protect the target from invasion by the vehicles. Each threat moves with

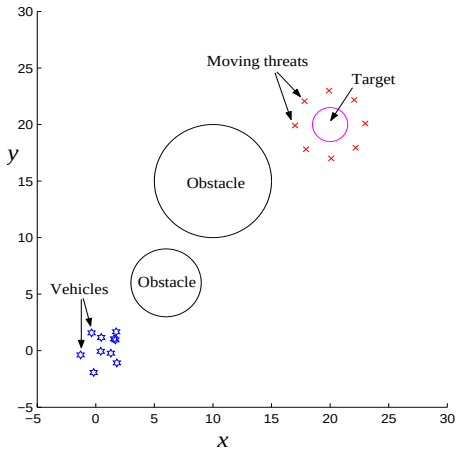


Figure 2.6: The simulation scenario.

angular velocity 0.03 rad/sec and radius 3 (linear speed 0.09/sec), while each vehicle's maximum speed is 0.06/sec. There is no stationary threat in the field. The optimal inter-vehicle distance r_0 is 0.5, the communication range $R_c = \frac{\sqrt{3}}{2}$, the detection range R_d for moving threats is 3, and the killing range $R_e = 0.5$. If a vehicle is inside the target area, it's motion is not affected by the threats and the obstacles. To guarantee the vehicles are distributed around the target center after they successfully enter the target area, an additional attractive potential from the target center is also included. The simulation was conducted in Matlab, where the function "fmincon" was used to solve the constrained minimization problem for each vehicle.

Fig. 2.7 shows the effect of the weighting constant λ_m for the potential due to moving threats. Other weights are fixed for Fig. 2.7(a) through (d): $\lambda_g = 1000$, $\lambda_o = 200$, and $\lambda_n = 1000$. For $\lambda_m = 10$ (very small), the vehicles paid least attention to the threats and four of them were destroyed because of getting too close to the threats (Fig. 2.7(a)); for $\lambda_m = 50$, only one vehicle was destroyed while the others entered the

target (Fig. 2.7(b)); for $\lambda_m = 200$, all vehicles entered the target successfully and in a timely manner (Fig. 2.7(c)); finally, for $\lambda_m = 2000$, some vehicles were not able to enter the target because more attention was put on evasion from the threats (Fig. 2.7(d)). We note that in all cases, the vehicles inside the target area displayed certain formations.

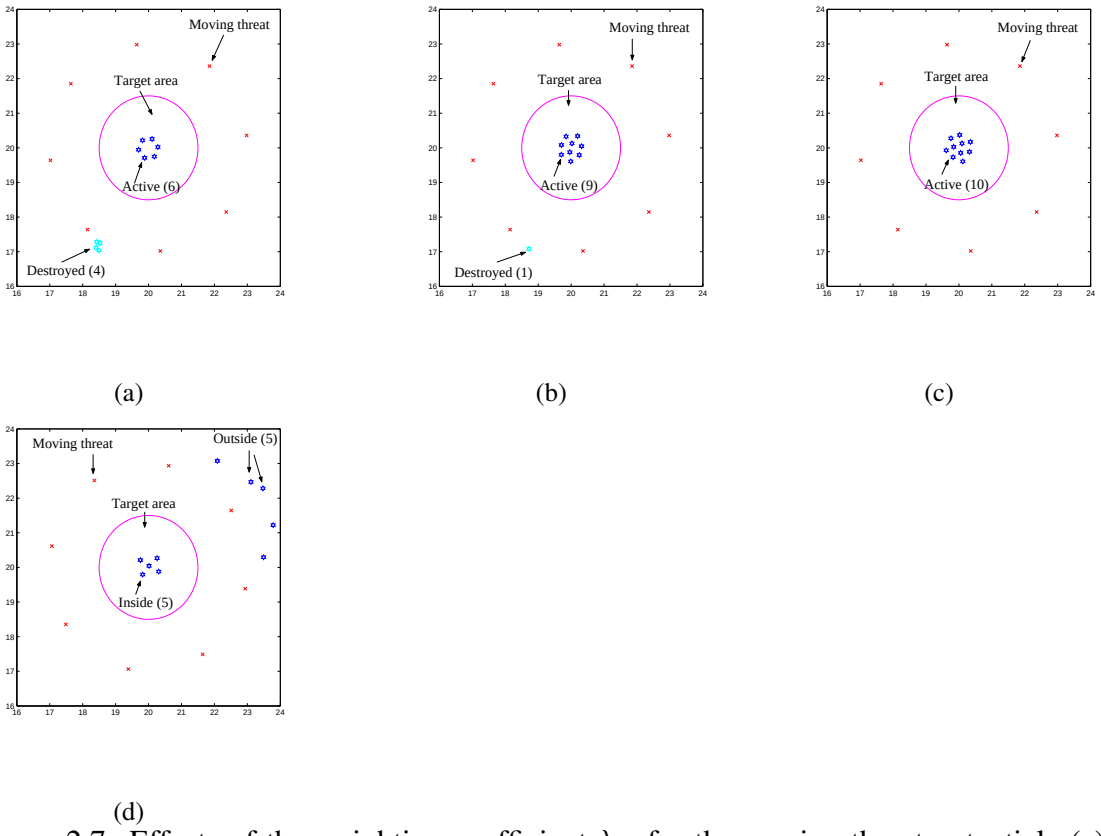
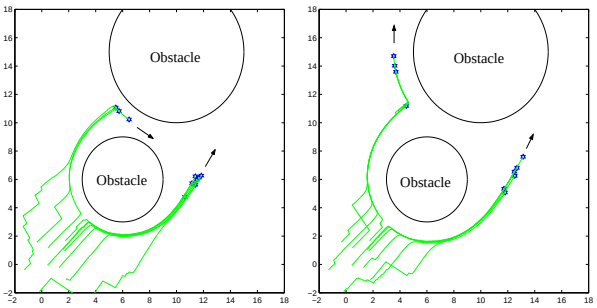


Figure 2.7: Effects of the weighting coefficient λ_m for the moving threat potential. (a) $\lambda_m = 10$; (b) $\lambda_m = 50$; (c) $\lambda_m = 200$; (d) $\lambda_m = 2000$.

Fig. 2.8 demonstrates the effect of the weighting constant λ_o for the obstacle potential. Other weighting constants used are: $\lambda_g = 50$, $\lambda_n = 200$, $\lambda_m = 200$. When $\lambda_o = 1000$, one group of vehicles took the shorter path to pass the obstacle valley (Fig. 2.8(a)); but when $\lambda_o = 5000$, no vehicle took the shortcut (some actually took the detour), as shown in Fig. 2.8(b).

From the simulation results, we see that the decentralized approach based on po-



(a) (b)

Figure 2.8: Effects of the weighting coefficient λ_o for the obstacle potential. (a) $\lambda_o = 1000$; (b) $\lambda_o = 5000$.

tential functions lead to some emergent behaviors of vehicles. In addition, we can modify the behaviors by appropriately changing the weighting constants.

2.4 Flocking and consensus problems

The application of the potential method is not limited to kinematic planning. We now consider the coordinated motion of a group of agents, whose goal is to converge to move in the same direction. This objective is referred to in the literature as flocking (e.g. [109, 83, 26]) due to the direct resemblance to flocking behavior of birds.

To meet this end, we change the neighboring potential in a way that suits our goal. Let $\theta_i(t)$ denote the heading of agent i at time t. We use the function $f_n(\|\theta_i - \theta_j(t)\|) = \frac{1}{2}\omega_{ij}\|\theta_i(t) - \theta_j(t)\|^2$, where the weights w_{ij} are such that:

$$\omega_{ij} = \omega_{ji} \geq 0$$

$$\omega_{ij} = 0 \Leftrightarrow j \nsim i.$$

For the ith agent, the neighboring potential is given by:

$$J_{i,t}^{n}(\theta_i) = \frac{1}{2}\sum_{i=1}^{N} \omega_{ij}\|\theta_i(t) - \theta_j(t)\|^2,$$

and the overall flock energy is:

$$J = \frac{1}{2}\sum_{i=1}^{N}\sum_{j\sim i} \omega_{ij}\|\theta_i(t) - \theta_j(t)\|^2.$$

The system equation based on the gradient method (2.2) is:

$$\dot{\theta}_i = -\sum_{j=1}^{N} \omega_{ij}(\theta_i(t) - \theta_j(t)) \tag{2.10}$$

Consider the (fixed, undirected) graph, $\mathcal{G}$ which models the agents and the links between them. Each node of the graph corresponds to an existing agent and there is a bidirectional link between any two agents whose distance is less than R_c. Let Q be the $N \times N$ matrix whose entries are given by:

$$q_{ij} = \begin{cases} \omega_{ij} & i \neq j \\ -\sum_{j\neq i} \omega_{ij} & i = j \end{cases}.$$

The matrix Q is known as the weighted Laplacian matrix for the graph $\mathcal{G}$, for the set of weights $\{\omega_{ij}\}$. The equation (2.10) can be rewritten in matrix form using Q as:

$$\dot{\theta} = -Q\theta. \tag{2.11}$$

The energy function J can also be written as in the matrix form

$$J = \frac{1}{2}\theta^T Q \theta.$$

The following properties of Q are well known.

Theorem 2.4.1. *The following hold:*

1. *Q is symmetric positive semidefinite.*

2. *If $\mathcal{G}$ is connected, Q has only one zero eigenvalue, $\lambda_n = 0$. This eigenvalue corresponds to the eigenvector $v_n = [11...1]^T$.*

3. *If $\mathcal{G}$ is not connected the multiplicity of eigenvalue $\lambda_n = 0$ is equal to the number of sets of mutually connected nodes in $\mathcal{G}$.*

Proof. See [77]. □

Corollary 2.4.2. *If the underlying communication graph is connected, the agents' heading will converge to the average of the agents' initial heading.*

Proof. The proof follows in a straightforward way from Theorem 2.4.1, and the fact that $\sum_{i=1}^{N} \theta_i(t)$ is preserved under equation 2.10. □

The equation 2.11 is known as (continuous) average consensus protocol. Olfati-Saber and Murray [86] consider average consensus problems in the presence of switching, time delays and asymmetrical communication. They have shown that in case of switching graphs

$$\|\delta(t)\|^2 = \|\theta(t) - \frac{\mathbf{1}^T\theta(0)}{N}\|^2$$

is a common Lyapunov function for the switching system

$$\dot{\theta} = -Q_{\sigma(t)}\theta, \tag{2.12}$$

where $\sigma(t)$ is a switching function that takes its value in an index set corresponding to the possible communication graphs, i.e. if $\mathcal{G} = \{\mathcal{G}_1, \mathcal{G}_2, ..., \mathcal{G}_s\}$ shows the set of possible graphs and $S = \{1, 2, ..., s\}$, then $\sigma(t) : R^+ \rightarrow S$ maps each point of time to a valid switching outcome.

As a special case, in the case of an unweighted graph, the weighted Laplacian simplifies to the regular Laplacian of the graph

$$L = D - A,$$

where A is the adjacency matrix of graph $\mathcal{G}$, and $D = \text{diag}(d_1, ..., d_N)$ is the diagonal matrix with degree of nodes on its diagonal. The Laplacian matrix L has spectrum

$$\lambda_1 \geq ...\lambda_{n-1} \geq \lambda_n = 0.$$

The statements of theorem 2.4.1 hold for the Laplacian L. In particular, if the graph is connected λ_n is a single eigenvalue.

In this case, λ_{n-1}, the second smallest eigenvalue of the Laplacian determines the speed of convergence of the fixed topology continuous consensus problem [86]. This eigenvalue is also referred to as the Fiedler eigenvalue or the algebraic connectivity. [1] The Fiedler eigenvalue is given by:

[1] The ordering of the eigenvalues of the Laplacian is usually done in reverse order to ours and the Fiedler eigenvalue is denoted by λ_2. We do not follow this convention to be consistent with our notations in the subsequent chapters

$$\lambda_{n-1}(L) = \min_{x \perp 1, x \neq 0} \frac{x^T L x}{||x||^2} \tag{2.13}$$

Since the disagreement vector $\delta(t) = \theta(t) - \frac{\mathbf{1}^T \theta(0)}{N}$ satisfies the equation $\dot{\theta} = -L\theta$, it can be easily shown that $V(\delta) = \frac{1}{2}||\delta||^2$ is a Lyapunov function for the disagreement dynamics and the Fiedler eigenvalue determines the speed of convergence [86] to consensus.

In the above model the effect of agents' position on their connectivity is not considered. In other words, it is assumed that two nodes that are connected at time t, remain connected regardless of their changing distance. Also in the switched model of [86] it is assumed that the switching dynamics is not dependent on the actual position of the agents. This problem can be addressed for example by letting the weights ω_{ij} be a decreasing function of the distance between p_i and p_j. In [26], Cucker and Smale proposed weights $\omega_{ij} = \eta(\|p_i - p_j\|^2)$, where $\eta(y) = \frac{K}{(\sigma^2 + y)^\beta}$ for fixed $\sigma, K > 0$ and $\beta > 0$.

They consider the system

$$\dot{p} = v$$

$$\dot{v} = -(Q \otimes I)v,$$

and study the convergence of the velocity vectors as a function of the initial positions and "loss" in communication modelled by β,k, and σ.

A cut-off function can be used to determine weights, e.g.,

$$\omega_{ij} = \begin{cases} 1 & \text{if} \quad \|p_i - p_j\| < R_c \\ 0 & \text{otherwise,} \end{cases}$$

which will result in Regular Laplacian (Q=L), where $L = D - A$ is the un-weighted Laplacian of the graph, which was discussed earlier.

The discrete-time version of flocking has been inspired by Vicsek's model for coordinated motion [117]. Vicsek's model assumes that all of the agents update their heading angles toward the average of their neighbors' heading angles. Based on Vicsek's model, Jadbabaie et al. [57] consider the model

$$\theta_i(t+1) = \frac{1}{n_i(t)+1} \sum_{j \in \mathcal{N}_i(t)} \theta_j(t), \qquad i = 1, 2, ..., N \tag{2.14}$$

Using graph dependent matrices A and D, this equation system can be written in vector notation as follows:

$$\theta(t+1) = (I + D(t))^{-1}(A(t) + I)\theta(t), \tag{2.15}$$

where $\theta(t) = [\theta_1(t), \theta_2(t), ..., \theta_N(t)]^T$.

Discretizing the equation (2.11) in time results in another set of weights that can be used to address flocking in discrete time. The resulting equation is

$$\theta(k+1) = (I - hQ)\theta(t), \tag{2.16}$$

in which the discritization constant h is chosen, such that all of the entries of $I - hQ$ remain non-negative.

The matrices $F_1 = (I + D)^{-1}(A + I)$ and $F_2 = I - hQ$ are stochastic, i.e. they are element-wise non-negative and $F_1\mathbf{1}_N = F_2\mathbf{1}_N = \mathbf{1}_N$. The convergence of discrete consensus schemes is addressed in the next chapter.

2.5 Invariant nodes and robustness of consensus problems

When designing a distributed dynamic system or algorithm, it is important to consider the change in the performance of the algorithm, when some of the agents do not follow the designed protocol, change their normal behavior, or simply fail. In this section, we study the robustness of the consensus protocol to such changes. We assume that a group of agents, labelled as "invariant nodes" cease to follow the consensus protocol and instead set their states to arbitrary fixed values. We study the effect of invariant nodes in this perspective, bearing in mind that these nodes can be considered as local leaders, adversaries, or intruders based on applications. We will later use the ideas developed in this section to propose a hierarchical network design architecture in Chapter 5.

2.5.1 Model

We consider a group of dynamic agents with some connectivity pattern modelled by a graph

$$\mathcal{G} = (\mathcal{V}, \mathcal{E}).$$

The vertex set $\mathcal{V} = \{1, 2, ..., n\}$ denotes the agents each given an arbitrary order. The edge set $\mathcal{E} = \{(i, j) | i \in \mathcal{V}, j \in \mathcal{V}, i \sim j\}$ represents the connectivity between the nodes. The relation $\sim$ denotes the neighborhood relation between the agents. To each agent i a state vector $\mathbf{x}_i \in R^m$ is corresponded. Without loss of generality, we assume that $\mathbf{x}_i = x_i \in R$ has scalar value. The generalization to the case of $m-$dimensional state vectors can be readily done using the Kronecker product and does not affect our results, so we use the

scalar case for simplicity of exposition. Therefore, the vector

$$x = [x_1, x_2, ..., x_n]^T$$

represents the state of the system.

We consider the problem for the time invariant fixed topology case. We divide the agents into two main groups, regular nodes and invariant nodes. Regular nodes are the agents who follow the designed consensus dynamics, i.e. they are considered to be integrators and the rate of the change of their state variables is determined by the difference of their state variable from those of their neighbors. The set D denotes the connected subset of the nodes which represent regular nodes. Invariant nodes are the agents who do not follow the regular dynamics. We assume that the value of each of the invariant nodes is fixed throughout all the time. We denote the set of invariant nodes by ∂D. The sets D and ∂D partition the set of vertices:

$$D \cup \partial D = \mathcal{V},$$

$$D \cap \partial D = \varnothing.$$

The problem is to study whether the system is stable in the presence of the invariant nodes and if so to what value will the regular nodes converge. Without loss of generality, we will assume that the regular nodes are labelled from 1 to N_r, i.e. $D = \{1, 2, ..., N_R\}$ and $\partial D = \{n_R + 1, n_R + 2, ..., n\}$. We denote by N_I the number of invariant nodes.

Consider the subgraph of $\mathcal{G}$ induced by the regular nodes. We denote this subgraph by $\mathcal{G}_R = (\mathcal{D}, \mathcal{E}_R)$, where $\mathcal{E}_R$ consists of all the edges between the regular nodes. We denote the adjacency matrix and the Laplacian of this graph respectively by A_R and L_R.

The dynamics of the regular nodes can be written as:

$$\dot{x_i} = \sum_{j \sim i} \left(x_j(t) - x_i(t) \right), \quad i \in D, \tag{2.17}$$

For the invariant nodes:

$$\dot{x_i} = 0 \Leftrightarrow x_i = \phi_i, \quad i \in \partial D, \tag{2.18}$$

where $\phi_{n_{R+1}}, ..., \phi_n$ are real constant values. The equations for regular nodes, 2.17 can be rewritten as:

$$\dot{x} = \sum_{j \sim i, j \in D} (x_i - x_j) + \sum_{j \sim i, j \in \partial D} (\phi_j - x_i), \quad i \in D$$

If we denote the number of invariant nodes which are neighbor to the regular node i by $n_i \geq 0$, the above equation is equivalent to:

$$\dot{x} = -n_i x_i + \sum_{j \sim i, j \in D} (x_i - x_j) + \sum_{j \sim i, j \in \partial D} \phi_j, \quad i \in D$$

This can readily be put into matrix form. Let M be the $n_R \times n_R$ matrix whose ith entry is the number of neighboring invariant nodes, n_i. The input of the system is determined by the state values of invariant nodes, i.e. $u = [\phi_{N_R+1}, ..., \phi_N]^T \in R^{N_I}$. The connectivity of invariant nodes to regular nodes is captured by a matrix $B \in R^{N_R \times N_I}$ whose entries B_{ij} are either 1 or 0 depending on whether the ith regular node is connected to the jth invariant node or not. Therefore the matrix representation of the equation 2.17 in our notation is:

$$\dot{x}_R = -\left(M + L_R \right) x_R + Bu, \tag{2.19}$$

where $x_R = [x_1, x_2, ..., x_{N_R}]^T$.

2.5.2 Stability

In this section we provide the results for stability and convergence of regular nodes and illustrate these results with examples. The first theorem considers the stability issue.

Theorem 2.5.1. *Let $\mathcal{G}$ be a connected graph with n nodes, out of which $N_I > 0$ are invariant and N_R are regular. For each invariant node let equation 2.18 determine the node's dynamics and*

$$x_i(t) = x_i(0) = \phi_i \in \partial D, \tag{2.20}$$

and for each regular node let equation 2.17 govern the node's dynamics. The resulting system of equations

$$\begin{cases} \dot{x_R} = -(M + L_R)x_r + Bu \\ \\ y_R = x_R \end{cases} \tag{2.21}$$

is L_p-stable , $\forall p \in [1, \infty]$ and each regular node will reach a finite steady state value provided that the invariant nodes' values are finite.

Proof. We consider the graph $\mathcal{G}$ and its partition into regular and invariant nodes. The dynamic equations of the system in matrix form is given by equation 2.21 with initial values determined by the invariant nodes and given by 2.20. We consider the quadratic form

$$V(x) = x_R^T(-M - L_R)x_R = -x_R^T L_R x_R - x^T M x = -\sum_{i\sim j, i,j\in D}(x_i - x_j)^2 - \sum_{i=1}^{N_R} n_i x_i^2.$$

Since there is at least one node i for which $n_i > 0$, equating $V(x) = 0$ results in $x_1 = x_2 = ... = x_{N_R} = 0$. Therefore the matrix $-(M + R)$ is negative definite.

The output of the system 2.21 is given by:

$$y_R(t) = x_R(t) = e^{-(L_R+M)t}x_0 + \int_0^t e^{-(L_R+M)(t-\tau)}Bu(\tau)d\tau$$

Note that $u(\tau) = u$ is time-independent. Since the matrix $-(M+L_R)$ is negative definite, the term $e^{-(L_R+M)t}$ can be exponentially bounded. Therefore for $t > 0$, and some positive constants k and a, we have

$$||y_R(t)|| \leq k_1 e^{-at} + k_2 \int_0^t e^{-a(t-\tau)}d\tau||u||,$$

where $k_1 = k||x_0||$ and $k_2 = k||B||$. Therefore, following [68], since $u \in \mathcal{L}_{pe}^{N_I}$, for all $p \in [1, \infty]$

$$||(y_R)_\tau||_{\mathcal{L}_p} \leq k_1\rho + \frac{k_2}{a}||u_\tau||_{\mathcal{L}_p},$$

in which

$$\rho = \begin{cases} 1 & p = \infty \\ (\frac{1}{ap})^{\frac{1}{p}} & p \in [1, \infty) \end{cases}$$

The system is finite gain $\mathcal{L}_p$ stable for all $p \in [1, \infty]$ with a bias term proportional to $||x_0||$.

Furthermore since

$$x_R(t) = e^{-(L_R+M)t}x_0 + \int_0^t e^{-(L_R+M)t(t-\tau)}Bu(\tau)d\tau,$$

and $-(L_R + M)$ is strictly negative definite, as $t \to \infty$, the term $e^{-(L_R+M)t}x_0$ will vanish and the term $\int_0^t e^{-(L_R+M)t(t-\tau)}Bu(\tau)d\tau$ will converge to a constant matrix. So the state of an arbitrary node i will converge to a finite value $x_i(\infty) < \infty$. □

Remark 2.5.1. *Since the invariant nodes' dynamics does not affect other invariant nodes, having edges between two invariant node is redundant. Therefore, in the formulation of*

the problem, it is implicitly assumed that either the subgraph induced by the regular nodes is connected or the graph consists of connected subgraphs of regular nodes separated by at most one invariant node.

If the graph $\mathcal{G}$ is not connected, it can be divided into connected subgraphs $\{\mathcal{G}_i = (\mathcal{V}_i, \mathcal{E}_i)\}_{i=1}^{K}$, where $K \leq n$. These graphs are such that their vertex sets partition $\mathcal{V}$ and their edge sets partition $\mathcal{E}$. Therefore the graph will be broken into connected subgraphs. We may have three different possible sets of combination of regular and invariant nodes in each subgraph, i.e. we may have:

- Subgraphs containing only invariant nodes: As we mentioned in the above remark these may be considered as isolated nodes
- Subgraphs containing only regular nodes: The nodes in such subgraphs will solve a average-consensus problem and each node's state variable converges to the average of the initial state of the nodes in that subgraph.
- Subgraphs containing both invariant and regular nodes: The stability of these subgraphs is covered by Theorem 2.5.1.

2.5.3 Convergence and relations to PDEs on graphs

In theorem 2.5.1 it was shown that in the presence of invariant nodes the state of regular nodes converges to a finite value provided that the invariant nodes have finite values. Now the objective is to determine these limits. With a small abuse of notation, suppose the steady state value of node i is denote by x_i. At the steady state the following equations should hold:

- For regular nodes:

$$\sum_{j\sim i}(x_i - x_j) = 0, \tag{2.22}$$

- For invariant nodes:

$$x_i = \phi_i. \tag{2.23}$$

Using the L, the Laplacian of the graph G equation 2.22 can be represented as:

$$(Lx)_i = 0 \quad i = 1, 2, ..., N_R. \tag{2.24}$$

By equation 2.24, the first N_R elements of the vector Lx are set to be zero. The remaining N_I entries are irrelevant. If we denote the number of node i's neighbors (regular and invariant) by n_i^*, equation 2.22 can be rewritten as:

$$x_i = \frac{1}{n_i^* + 1}\left(x_i + \sum_{j\sim i} x_j\right) \quad i = 1, 2, ..., N_R. \tag{2.25}$$

If we let $P = (I + D)^{-1}(A + I)$, where A is the adjacency matrix for graph $\mathcal{G}$, and D is the diagonal matrix with ith diagonal equal to n_i^*, then equation 2.25 can be cast as:

$$x_i = (Px)_i = \left[\left((I + D)^{-1}(A + I)\right)x\right]_i \quad i = 1, 2, ..., N_R. \tag{2.26}$$

Note that P is a stochastic matrix and equation 2.26 is valid for the first N_R entries of X_i.

Consider a random walk $\{X_n\}_{n\geq 0}$ on the graph $\mathcal{G}$ with respect to the transition matrix P. This is a discrete Dirichlet problem and in theorem 2.5.2 we show that the values to which the regular state variables converge can be determined as hitting times of this random walks. The proof is standard and we essentially follow the proof of ([18], Theorem 5.2.1). Let the random walk start at a regular node $i \in D$ and continue in the

subgraph D induced by the regular nodes until it hits the boundary set ∂D of the invariant nodes. Define the function

$$\begin{aligned} \phi: \quad & \partial D \to R \\ & \phi(i) = \phi_i \end{aligned} \tag{2.27}$$

Let T be the hitting time of ∂D and define the function

$$\begin{aligned} h: \quad & V \to R \\ h(i) = \quad & E[\phi(Z_T)|Z_0 = i] \quad i \in D \\ & \phi(i) \quad i \in \partial D \end{aligned} \tag{2.28}$$

We can now state:

Theorem 2.5.2. *If the graph $\mathcal{G}$ is connected, in the steady state the state x of system (2.17) , (2.18) can be determined uniquely and converges to a vector h, entries of which is defined by (2.28).*

Proof. Since the underlying graph $\mathcal{G}$ is connected, P is irreducible. Also $\forall i \in P\ p_{ii} > 0$, which means that the chain is also aperiodic. The number of states is finite and therefore the chain is positive recurrent and $P(T < \infty | Z_0 = i) = 1$. It was shown that

$$x = \begin{cases} Px & \text{on } D \\ \phi & \text{on } \partial D \end{cases} \tag{2.29}$$

The entries of h on ∂D are defined as constant values. On D, first step analysis yields:

$$h(i) = \sum_{j \sim i} p_{ij} h(j).$$

So h satisfies the equation 2.29 and $h = x$ on $\mathcal{V}$.

If u is another solution to equation 2.29, then following [18]

$$M_n = u(X_n) - u(X_0) - \sum_{k=0}^{n-1}(P - I)u(X_k)$$

is a Levy Martingle with respect to $\{X_n\}_{n\geq 0}$. Let $M_{T\wedge k}$ denote process M_k stopped at T. Then optional sampling theorem yields: $E[M_{T\wedge K|X_0=i} = E[M_0|X_0 = i]$ for any integer $K > 0$. Therefore by using the condition 2.29 and dominated convergence, we get

$$u(i) = E[u(X_{T\wedge K})]|X_0 = i] = E[u(X_T)|X_0 = i] = h(i).$$

since ϕ is bounded and $P(T < \infty|Z_0 = i) = 1$ the function h is bounded and therefore for the regular nodes in the system $(2.17), (2.18)$, the unique bounded solution is:

$$\lim_{t\to\infty} x_i(t) = E[\phi(X_T)|X_0 = i].$$

□

We illustrate the above results with a simple example. Consider the network of dynamic agents given by figure 2.9, where $\mathcal{V} = \{1, 2, 3, 4, 5\}$, $D = \{1, 2, 3\}$, and $\partial D = \{4, 5\}$. The matrices L, L_R, M, B, P are given by:

$$L = \begin{pmatrix} 2 & -1 & -1 & 0 & 0 \\ -1 & 3 & -1 & 0 & -1 \\ -1 & -1 & 3 & -1 & 0 \\ 0 & 0 & -1 & 1 & 0 \\ 0 & -1 & 0 & 0 & 1 \end{pmatrix}$$

$$P = \begin{pmatrix} 1/3 & 1/3 & 1/3 & 0 & 0 \\ 1/4 & 1/4 & 1/4 & 0 & 1/4 \\ 1/4 & 1/4 & 1/4 & 1/4 & 0 \\ 0 & 0 & 1/2 & 1/2 & 0 \\ 0 & 1/2 & 0 & 0 & 1/2 \end{pmatrix}$$

$$L_R = \begin{pmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{pmatrix}; M = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \text{ and}$$

$$B = \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}$$

The differential equations of the system evolution can be written as:

$$\dot{x}_R = -(M + L_R)x_R + Bu \Rightarrow$$

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{pmatrix} = - \left[\begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} + \begin{pmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{pmatrix} \right] \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

$$+ \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \phi_4 \\ \phi_5 \end{pmatrix} \Rightarrow$$

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{pmatrix} = \begin{pmatrix} -2 & 1 & 1 \\ 1 & -3 & 1 \\ 1 & 1 & -3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} + \begin{pmatrix} 0 \\ \phi_5 \\ \phi_4 \end{pmatrix}$$

At the steady state for $i \in D$:

$$\left(\begin{array}{ccccc} 2 & -1 & -1 & 0 & 0 \\ -1 & 3 & -1 & 0 & -1 \\ -1 & -1 & 3 & -1 & 0 \\ \hline 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{array}\right) \left(\begin{array}{c} x_1 \\ x_2 \\ x_3 \\ \hline x_4 \\ x_5 \end{array}\right) = \left(\begin{array}{c} 0 \\ 0 \\ 0 \\ \hline \phi_4 \\ \phi_5 \end{array}\right) \tag{2.30}$$

Also for $i \in D = \{1, 2, 3\}$, $(Lx)_i = 0$, or equivalently $(Px)_i = x_i$. For $i \in \partial D = \{4, 5\}$:

$$x_i = \phi_i \Rightarrow \begin{pmatrix} x_3 \\ \\ x_4 \end{pmatrix} = \begin{pmatrix} \phi_4 \\ \\ \phi_5 \end{pmatrix} = \phi$$

So overall:

$$x = \begin{cases} Px \quad on \quad D \\ \\ \phi \qquad \partial D \end{cases} \Rightarrow x_i = E[\phi(Z_T)|Z_0 = i] \tag{2.31}$$

Solving the system of equations (2.30) yields:

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} = \begin{pmatrix} \frac{\phi_4+\phi 5}{2} \\ \frac{3\phi_4+5\phi_5}{8} \\ \frac{5\phi_4+3\phi_5}{8} \\ \phi_4 \\ \phi_5 \end{pmatrix} \tag{2.32}$$

The values of x_i's can be also obtained from equation (2.31) as will be shown in following.

For $i \in D$ and $j \in \partial D$, let p_{ij} denote $P[Z_T = j | Z_0 = i]$. Then:

$$\begin{cases} p_{15} = \frac{1}{2}p_{25} + \frac{1}{2}p_{35} \\ p_{25} = \frac{1}{3}p_{15} + \frac{1}{3}p_{35} + \frac{1}{3} \\ p_{35} = \frac{1}{3}p_{15} + \frac{1}{3}p_{25} \end{cases} \Rightarrow$$

$$\begin{pmatrix} p_{15} \\ p_{25} \\ p_{35} \end{pmatrix} = \begin{pmatrix} 1/2 \\ 5/8 \\ 3/8 \end{pmatrix} \text{ and } \begin{pmatrix} p_{14} \\ p_{24} \\ p_{34} \end{pmatrix} = \begin{pmatrix} 1/2 \\ 3/8 \\ 5/8 \end{pmatrix}$$

Therefore:

$$E[\phi(Z_T | Z_0 = i) = E[\phi(Z_T) | Z_0 = i, Z_T = 4] P[Z_T = 4 | Z_0 = i]$$

$$+E[\phi(Z_T) | Z_0 = i, Z_T = 5] P[Z_T = 5 | Z_0 = i] \Rightarrow$$

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} = \begin{pmatrix} \frac{\phi_4 + \phi 5}{2} \\ \frac{3\phi_4 + 5\phi_5}{8} \\ \frac{5\phi_4 + 3\phi_5}{8} \\ \phi_4 \\ \phi_5 \end{pmatrix}$$

which is the same as result given by (2.32).

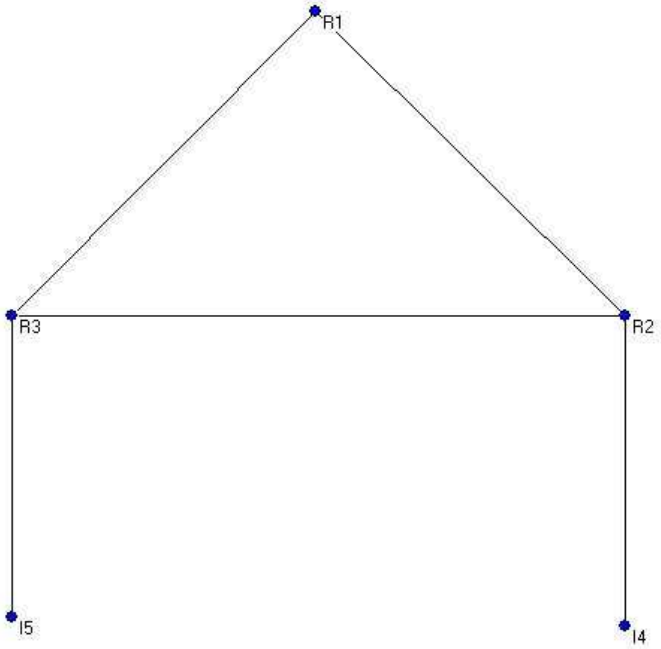


Figure 2.9: Simple illustration of invariant nodes' effect

Chapter 3

Convergence in consensus problems and the implications for efficiency of distributed algorithms

3.1 Notation and framework

Consider a set of n agents and model the interconnection between them by a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. The nodes of the graph, $\mathcal{V} = \{1, 2, ..., n\}$ represent the agents. The graph can be directed due to possible asymmetry in communication: if node i is able to communicate with node j, a directed edge exists from i to j. In this case we say that node i is a *neighbor* of node j. In general the interaction graph can be time-varying, i.e. the agents' neighbors may change over time. Each agent is trivially a neighbor to itself. The set of neighbors of node i is denoted by N_i. An undirected graph represents bi-directional communication: if node i is able to communicate with agent j at time t, then agent j is also able to communicate with agent i. A fixed graph means that there is no change in neighbors of any of the agents over time.

Consider that any agent i maintains a variable x_i, its subjective evaluation of a *coordination variable* $\hat{x}$.[1] A discrete time consensus equation is meant to be a linear

[1]The coordination variable can be a vector. However, we consider a scalar valued $\hat{x}$ for ease of exposition. The generalization to vectors is straightforward using Kronecker products. The convergence results for the vector case follow from the scalar case, so we will limit the exposition to the case of scalar coordination variable.

iteration of the following form: each agent i starts with a value $x_i(0)$, and updates it value at time t as a convex combination of the values of its neighbors according to equation:

$$x_i(t+1) = f_{ii}(t)x_i(t) + \sum_{j \in N(i)} f_{ij}(t)x_j(t) \tag{3.1}$$

in which

$$\forall i \in \{1, ..., n\} : \qquad \sum_j f_{ij} = 1$$

and

$$\forall i, j \in \{1, 2, ..., n\} : \qquad f_{ij} \geq 0$$

$$\forall i \in \{1, ..., n\} : \qquad f_{ii} \geq \alpha > 0$$

Equation(3.1) can be written in matrix form:

$$x(t+1) = F(t)x(t) \tag{3.2}$$

where $F(t)$ is an aperiodic stochastic matrix.

Let $\mathbb{G} = \{\mathcal{G}_0, ..., \mathcal{G}_{M-1}\}$ be the finite set of all possible interaction graphs defined by a set of agents. For any interaction graph $\mathcal{G}_k$ consider a predetermined rule that generates the weights that each agent applies to its neighbors. Therefore, we assume that for each graph topology $\mathcal{G}_k$, there is a corresponding weight matrix F_k. The entries of each F_k are constrained as in equation (3.1) and $F_k(i, j)$ is nonzero if and only if the agent j is a neighbor of the agent i. Let $\mathbb{F} = \{F_0, ...F_{M-1}\}$ be the set of (weight) matrices corresponding to the set of interaction graphs $\mathbb{G}$. Let $\mathbb{M} = \{0, ..., M-1\}$ be a set of indices. Consider a switching signal $\sigma : \mathbb{N} \bigcup \{0\} \rightarrow \mathbb{M}$. Then equation (3.1) can be written as a switched system:

$$x(t+1) = F_{\sigma(t)}x(t). \tag{3.3}$$

The discrete time consensus framework studies whether using a linear consensus scheme, the group is able to reach an agreement regarding the coordination value. Extensions to asynchronous and nonlinear protocols can also be studied.

Definition 3.1.1. *Consider n agents each maintaining a coordination variable x_i. The group is said to reach a consensus asymptotically, if starting from any initial condition $x(0) = [x_1(0), ..., x_n(0)]^T$ and using update equations (3.1) , $\forall i, j \in \{1, ..., n\}$, $||x_i(t) - x_j(t)|| \to 0$ as $t \to \infty$.*

Since $x(t) = \lim_{N\to\infty} F(N)...F(2)F(1)F(0)x(0)$, the above definition is equivalent with the set of matrices $\mathbb{F}$ being weakly ergodic.

A stronger version of definition (3.1.1) is the following:

Definition 3.1.2. *Consider n agents each maintaining a coordination variable x_i. The group is said to reach a consensus asymptotically, if starting from any initial condition $x(0) = [x_1(0), ..., x_n(0)]^T$ and using update equations (3.1) , there exists a variable $\hat{x}$, such that $\forall i \in \{1, ..., n\}$, $||x_i(t) - \hat{x}|| \to 0$ as $t \to \infty$.*

Chatterjee and Seneta [22] show that for consensus problems weak and strong ergodicity are equivalent.

Theorem 3.1.1. *(Chatterjee and Seneta [22])*

For the products of the form $F(N)...F(2)F(1)F(0)$, weak and strong ergodicity are equaivalent.

Proof. It suffices to show that weak ergodicity implies strong ergodicity. Let

$$P^{(k)} = F(k)...F(1)F(0). \tag{3.4}$$

For $\epsilon > 0$, weak ergodicity implies:

$$-\epsilon \leq P_{j,m}^{(k)} - P_{i,m}^{(k)} \leq \epsilon$$

for some $k \geq k_0$, $\forall i, j, m \in \{1, 2, ..., n\}$. Since $P^{(k+1)} = F(k+1)P^{(k)}$,

$$\sum_{j=1}^{n} F_{hj}(k+1)(P_{i,m}(k) - \epsilon) \leq \sum_{j=1}^{n} F_{hj}(k+1)P_{j,m}(k) \leq \sum_{j=1}^{n} F_{hj}(k+1)(P_{i,m} + \epsilon).$$

Hence,

$$P_{i,m}(k+1) - \epsilon \leq P_{h,m}(k+1) \leq P_{i,m}(k) + \epsilon,$$

for $k \geq k_0$. Therefore, it follows that for any $q \geq 1$, and $i, j, m \in \{1, ..., n\}$, there exists k_0, such that for $k \geq k_0$:

$$|P_{j,m}^{k+q} - P_{i,m}^{k}| \leq \epsilon.$$

If we let $j = i$, the result follows from Cauchy convergence of the sequence $P_{i,m}^{(k)}$, as $k \to \infty$. □

3.2 Review of convergence Results for consensus problems

A set of matrices $\Sigma = \{F_1, ...F_m\}$is called Left Convergent Product (LCP) if any left infinite product (in the form of equation (3.4)) of the matrices from the set converge to a limit. The convergence of consensus iterations is therefore related to whether contiguous sequences of agent configurations corresponds to sets of weight matrices with LCP property. The interesting issue is to find the relationship between the algebraic conditions for convergence of the product (3.4) and the topology of the interaction graphs. In this section, we review the existing results on the convergence of consensus iterations. The mathematical tools for studying this problem are mainly from the theory of non-negative

matrices and graph theory. A summary of the relevant terminology and results is included in the Appendix.

The main idea behind most of the results of this kind is as follows. Let $m(t) = min_i x_i(t)$ and $M(t) = max_i x_i(t)$. since each $F_{\sigma(t)}$ is stochastic, it is straightforward to prove that $m(t)$ is nondecreasing and $M(t)$ is nonincreasing. It is then sufficient to verify the difference $M(t) - m(t)$ is reduced by a constant factor over a sufficiently large time interval; the interval should be chosen in a way that every agent gets to influence every other agent or at least there is a set of agents that influence all nodes. In this way the above difference approaches zero and the desired consensus is reached. The convergence condition is thus satisfied when the agents can influence each other "sufficiently". Results on the convergence of consensus problems were first established by Tsitsiklis and coauthors in the context of distributed computing and detection [115, 114, 11]. Results with more graph theoretic emphasis were established in the context of flocking systems by several authors [57, 95, 79, 14, 36]. We provide a summarized review of the convergence results.

3.2.1 Convergence in symmetrical neighborhood case

Jadbabaie et al. [57] consider the convergence of the Vicsek model in which the neighborhood relation is symmetric. The symmetry is only with regard to connections. The weights may not be symmetric. In this model, we have:

$$x_i(t+1) = \frac{1}{1+n_i(t)}[x_i(t) + \sum_{j \in N_i(t)} x_j(t)] \tag{3.5}$$

Considering the set of all possible graphs on n vertices $\mathbb{G}$ and an indexing set $\mathbb{M}$, for each $p \in \mathbb{M}$ define weight matrix

$$F_p = (I + D_p)^{-1}(A_p + I) \tag{3.6}$$

where A_p is the adjacency matrix of the graph $\mathcal{G}_p$ and D_p is the diagonal matrix whose ith diagonal element is the degree of vertex i. Here, we have excluded each node from its own adjacency and diagonal matrix. Since F_p is a stochastic matrix, it can be thought of as a transition matrix of a Markov chain. So, to the system (3.5) we can correspond a Markov chain with time varying transition probabilities. Therefore, reaching consensus in system (3.6) is equivalent to reaching a steady state invariant probability distribution for the corresponding Markov chain. For a fixed interconnection topology, the graph should be connected for the corresponding Markov chain to be irreducible and aperiodic. Aperiodicity follows because of self loops at each node.

For a changing topology, the set $\mathbb{F}_c = \{F_0, F_2, ..., F_{Q-1}\}$, corresponding to the set of *connected* graphs is an LCP set. This means that if switching is occurred among the matrices corresponding to connected graphs, the convergence will occur with an exponential rate. However, it is possible to relax this condition as in [57].

The union of a set of graphs $\{\mathcal{G}_{i1}, \mathcal{G}_{i2}, ..., \mathcal{G}_{im}\}$ each with the vertex set $\mathcal{V}$, is a graph with the vertex set $\mathcal{V}$ and the edge set equal to the union of the edge sets of graphs. This set is considered *jointly connected* if the union of its members is a connected graph. Considering a dynamic graph process $G(t)$, for which at each time a graph from $\mathbb{G}$ is selected due to an underlying law. The n agents under consideration are defined to be *linked together across a time interval* $[t, \tau]$ if the set of graphs $\{G_{\sigma(t)}, G_{\sigma(t+1)}, ..., G_{\sigma(\tau)}\}$

encountered along the interval is jointly connected. A general sufficient condition for reaching consensus is stated below.

Theorem 3.2.1. *(From [57]) Let x_0 be fixed and let σ denote a switching signal for which there exists an infinite sequence of contiguous, non-empty, bounded, time-intervals $[t_i, t_{i+1})$ starting at t_0, with the property that across each such interval, the n agents are linked together. Then $\lim_{t\to\infty} x(t) = x_{ss}\mathbf{1}$ where x_{ss} is a number depending only on x_0 and σ.*

The basic idea behind proof of Theorem 3.2.1 is based on a classical theorem by Wolfowitz [120]:

Theorem 3.2.2. *Let $M_1, M_2, ...M_m$ be a finite set of matrices with the property that for each sequence $M_{i1}, M_{i2}, ..., M_{ij}$ of positive length, the matrix product $M_{ij}M_{ij-1}...M_{i1}$ is ergodic. Then for each infinite sequence, $M_{i1}, M_{i2}, ...$ there exists a column vector c such that*

$\lim_{j\to\infty} M_{ij}M_{ij-1}...M_{i1} = \mathbf{1}c^T$.

3.2.2 Convergence in asymmetrical neighborhood case

For the case of asymmetrical neighborhood, references [79, 20], [95], [14] and [36] provide sufficient conditions for convergence. The following definitions provide a framework for formalizing the reachability of the nodes, which is essential for convergence results [79].

Consider graphs on a vertex set $\mathcal{V} = \{1, ..., n\}$. The composition of two graphs $\mathcal{G}_2$ and $\mathcal{G}_1$, denoted by $\mathcal{G}_2 \circ \mathcal{G}_1$ is a directed graph with vertex set $\mathcal{V}$ and edge set defined in

such a way that (i, j) is an edge of the composition if there is a vertex q such that (i, q) is an edge of $\mathcal{G}_1$ and (q, j) is an edge of $\mathcal{G}_2$. A graph $\mathcal{G}$ is said to be rooted in node i (equivalently i is a root of graph $\mathcal{G}$) if any other vertex of $\mathcal{G}$ is reachable from i along a path in the graph. A rooted graph is a graph which has at least one root. A finite sequence of directed graphs is jointly rooted if the composition $\mathcal{G}_{p_k} \circ \mathcal{G}_{p_{k-1}} \circ ... \circ \mathcal{G}_{p_1}$ is rooted. An infinite sequence of graphs $G_{p1}, G_{p2}, ...$ is repeatedly jointly rooted if there is a positive integer m for which each finite sequence $\mathcal{G}_{p_{m(k-1)+1}}, ..., \mathcal{G}_{p_{mk}}, k > 1$ is jointly rooted. The following result of Morse [79] gives the sufficient condition for convergence of asymmetrical neighborhood case in terms of the property of being jointly rooted of the interaction graphs.

Theorem 3.2.3. *(from [79]) Let x_0 be fixed and let σ denote a switching signal for which the infinite sequence of graphs $G_{\sigma(0)}, G_{\sigma(1)}, ...$ is repeatedly jointly rooted. Then there is a constant steady state heading x_{ss}, depending only on x_0 and σ for which $\lim_{t\to\infty} x(t) = x_{ss}\mathbf{1}$. where the limit is approached exponentially fast.*

Other sufficient conditions with slightly different connectivity assumptions exists, since the reachability condition can be also stated in terms of spanning trees. The reference [95] a requires uniform spanning tree condition:

Theorem 3.2.4. *If there exists a nonnegative integer B such that $\cup_{s=t}^{t+B} G_{\sigma(s)}$ contains a spanning tree for all t, the same result holds.*

3.2.3 Convergence in asynchronous case

Iterations (3.1) and (3.3) are synchronous in the sense that all the agents update their states at the same time using the latest values of their neighbors. In the asynchronous setting, the order in which states are updated is not fixed and the frequency of updating is also arbitrary. If we denote the set of times there is an update in system by k, the set of updating agents at time k by $S(k)$ and iteration delay from agent j to agent i at time k by $d(i,j,k) \geq 0$, the dynamics of the asynchronous system can be written as:

$$x_i(k+1) = \begin{cases} \sum_j F_{ij}(k)x_j(k-d(i,j,k)) & if \quad i \in S(K) \\ x_i(k) & otherwise \end{cases} \tag{3.7}$$

Two assumptions are made. First, we need a bound on delays, i.e. $0 < D(i,j,k) < D$ for all agents involved. This is known as partial asynchronism in literature [11]. The second assumptions is that every agent is updated infinitely often, i.e $\bigcup_{k=T}^{\infty} S_k = \{1,2,...,n\}$ for any T.

The following result holds [36]

Theorem 3.2.5. *Let $G(t)$ be a time varying interaction graph at time t, with weights selected from a finite set of arbitrary positive numbers. Global consensus is reached asymptotically if and only if there exists an infinite sequence of contiguous bounded time intervals starting at t_0, with the property that across each such interval the union of the interaction graphs has a spanning tree.*

Other conditions for convergence of consensus problems under different connectivity and synchrony assumptions can be found in [37] and references therein.

3.3 Continuity in convergence and robustness issues for consensus problems

A direct result of the convergence properties of consensus problems is that the steady state value of consensus is a function of the initial state of agents x_0, the switching signal σ, and the set of weight matrices Σ.

$$x_{ss} = \sum_{i=1}^{n} c_i(\sigma, \Sigma) x_i(0)$$
$$\sum_{i=1}^{n} c_i(\sigma, \Sigma) = 1. \tag{3.8}$$

In this section, two problems concerning the value to which a consensus iteration converges and its robustness to changes in switching signals are reviewed. These two problems are:

1. Let $\epsilon > 0$, $c = [c_1, c_2, ..., c_n]^T$ be a given vector such that $0 \leq c_i \leq 1$, and $\mathbf{1}^T c = 1$. Is there any set of weight matrices $\Sigma = \{F_0, ...F_{m-1}\}$ and switching sequence σ, for which

$$||x_{ss} - c^T x_0|| \leq \epsilon?$$

2. For two arbitrary sequences $\sigma = (\sigma_1, \sigma_2, ...)$ and $\sigma' = (\sigma'_1, \sigma'_2, ...)$, denote by D, the set of time instances, for which the switching signals do not agree, i.e.

$$i \in D \Leftrightarrow \sigma_i \neq \sigma'_i.$$

For a given set of weight matrices (e.g. Vicesk weights), let x_{ss} and x'_{ss} respectively denote the consensus value if the sequences σ and σ' are used. What are the

conditions on D, so that for given $\epsilon > 0$,

$$||x_{ss} - x'_{ss}|| \leq \epsilon?$$

Both of these problems can be addressed using the continuity property of convergence in consensus problems. It should be noted however, that the solution to the first problem can be easily determined if one has control over the choice of weight matrices. In this case, a solution to the first problem can be provided using Metropolis weights [18].

Theorem 3.3.1. *The set of weights given by the following procedure solves problem 1, over a fixed connected topology:*

- *Take an arbitrary stochastic matrix K that satisfies the connectivity constraints. (equivalently, let each node assign arbitrary weights to its neighbors)*
- *For $i \sim j$, Let $R(i,j) = \frac{c_j K_{ji}}{c_i K_{ij}}$, then:*

$$F_{ij} = \begin{cases} K_{ij} \min(1, R(i,j), & \textit{if} \quad i \neq j \\ K_{ij} + \sum_k (1 - \min(1, R(i,j))), & \textit{else} \end{cases} \tag{3.9}$$

Proof. Since $c_i F_{ij} = c_j F_{ji}$, the matrix F is a stochastic matrix corresponding to a reversible Markov chain which accepts c as its stationary distribution. Therefore

$$\lim_{t \to \infty} F^t = \mathbf{1}c.$$

□

The main challenge is when the weights are determined by a given rule, e.g. Vicsek's weights $F = (I + D)^{-1}(A + I)$, and the effect of switching is studied. this is

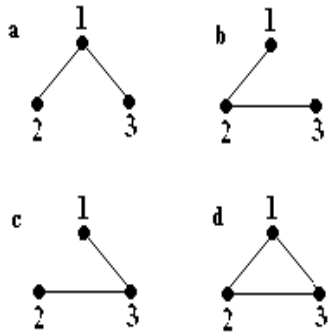


Figure 3.1: Example: Different connectivity graphs for 3 agents

closely related to the second problem. The following example illustrates some aspects of this problem.

Example Consider three agents in figure 3.1 whose state evolve according to $x(t+1) = (I + D(t))^{-1}(I + A(t))$.

Consider that they use one of four topologies $(a), (b), (c), (d)$. There are four possible formation matrices F_1, F_2, F_3, F_4 corresponding to these topologies:

$$F_1 = \begin{bmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} \end{bmatrix} F_2 = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ 0 & \frac{1}{2} & \frac{1}{2} \end{bmatrix} F_3 = \begin{bmatrix} \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{bmatrix} F_4 = \begin{bmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{bmatrix}$$

Suppose that as a design criterion only one of the above topologies, say (a) is desired, i.e. we want the agents to converge to a coordination value given by equation

$$x(t+1) = F_1 x(t) \tag{3.10}$$

A straight forward computation shows that in this case,

$$\lim_{t\to\infty} F_1^t = \begin{bmatrix} \frac{3}{7} & \frac{2}{7} & \frac{2}{7} \\ \frac{3}{7} & \frac{2}{7} & \frac{2}{7} \\ \frac{3}{7} & \frac{2}{7} & \frac{2}{7} \end{bmatrix}.$$

So all the agents are expected to converge to $x_{ss} = \frac{3}{7}x_1(0) + x7\theta_2(0) + \frac{2}{7}x_3(0)$. However, suppose that due to the agents' autonomy and communication failures or mistakes, the other topologies $(b, c,$or $d)$ are also likely to happen. An interesting issue is to obtain conditions under which the agents reach the consensus within an acceptable error bound. For example, consider a switching sequence

$$\sigma = \{\underbrace{..., 3,}_{\infty times} \quad \underbrace{2, ..., 2,}_{10 times} \quad \underbrace{1, ..., 1}_{10 times}\}$$

It is interesting to denote that $lim_{t\to\infty}||F_1^t - F_3^t F_2^{10} F_1^{10}||_2 \cong 1.7078 \times 10^{-4}$. Therefore difference between the actual and desired coordination variables is small, despite the fact that at most of the time the actual switching is different from the desired one. However, it is important to notice that at initial 10 steps, the switching signals are identical.

In general it can be seen that if the switching sequences σ and σ' have the same initial subsequence up to a sufficiently large time, i.e. $\sigma[i] = \sigma'[i]$ for $i = 1, ..., N$ where N is large enough then the infinite remainder of sequence in not significant enough and we will have $||x_{ss} - x'_{ss}|| < \epsilon$ for any desired ϵ. These observations are addressed in the following section. The analysis is based on the results from nonnegative matrix theory [29] [12] [53].

3.3.1 Continuity in convergence [53]

A set of matrices $\Sigma = \{F_0, ...F_{m-1}\}$is called Left Convergent Product (LCP) if any left infinite product of the matrices from the set converge to a limit. Σ is said to be para-contractive with respect to a vector norm $||.||$, if $\forall F_p \in \Sigma, x \in R^n$

$$F_p x \neq x \Rightarrow ||F_p x|| < ||x|| \tag{3.11}$$

Now consider the sequence space $S = \{\sigma : \sigma = (\sigma_1, \sigma_2, ...)\}$. where each $\sigma_i \in \{0, 1, ..., m-1\}$. If Σ is LCP, the product $...F_{\sigma-n}...F_1$ converges to a matrix A. Define a function ϕ

$$\phi : S \rightarrow M_n$$

$$\phi(\sigma) = A, \tag{3.12}$$

and a metric ∂ on S

$\partial(\sigma, \sigma') = m^{-k}$,

The index K is the first index such that $\sigma \neq \sigma'$. If function ϕ is continuous on S using the metric ∂ and any norm on M_n, then Σ is called a continuous LCP. More formally:

Definition 3.3.1. *(From [53]) ϕ is continuous at $\sigma \in S$ if given any $\epsilon > 0$ there is an integer K such that if $k > K$, then $||\phi(\sigma) - \phi(\sigma')|| < \epsilon$ for all σ' that differ from σ after the k-th digit.*

Therefore, saying that an LCP set is continuous is the same as saying that the product $...F_{\sigma_K}...F_1$ does not change much regardless of the choice of the matrices $F_{\sigma_{k+1}}, F_{\sigma_{k+2}}, ...$

and so on. Our goal is to show that the set Σ is a continuous LCP. For this we need another result. Suppose $E(F)$ denote the 1-eigenspace of matrix F. $E(f) = \{x|Fx = x\}$. Let $\Sigma = \{F_0, ..., F_{m-1}\}$ be an LCP set. Define $E(\Sigma) = E(F_{m-1}) \bigcap E(F_{m-2}) \bigcap ... E(F_0)$. The following theorem is a crucial fact in our argument.

Theorem 3.3.2. *(From [12]) Let $\Sigma = \{F_0, ..., F_1\}$. Then Σ is a continuous LCP set if and only if Σ is a paracontractive set and $E(\Sigma) = E(F_i)$ for all i.*

The main result in this section is the following theorem based on [29] and [53]:

Theorem 3.3.3. *In an undirected neighborhood system with switching topology if the agents remain connected all the time, their convergence to consensus is continuous. Furthermore, if the switching signal is such that there exists an infinite sequence of contiguous, non-empty, bounded, time-intervals $[t_i, t_{i+1})$ starting at t_0, with the property that across each such interval, the n agents are linked together, the result will hold.*

Sketch of Proof Let $\Sigma = \{F_1, ..., F_{m-1}\}$ denote the set of all valid F matrices corresponding to connected topologies G_c. It was shown that this set is LCP. The continuity of this LCP is a result of theorems 3.3.2, the following theorem 3.3.4 and the fact that $E(\Sigma) = \mathbf{1}$. The extension to graphs linked through time is straightforward by defining a new LCP set based on the matrices corresponding to the combined graphs. Since there is an upper bound T on the length of the contiguous intervals, it can be shown that the product of F matrices of length at most T is an LCP set and the result follows.

Theorem 3.3.4. *There exists a norm $||.||$ such the set $\mathbb{F}$ is paracontractive with respect to that norm.*

Sketch of Proof Since $\mathbb{F}$ is LCP, we have that $\lim_{t\to\infty} ||x_{t+1} - xt|| = 0$. It can be shown ([53]) that this implies that there exist a constant L such that $\sum_{t=1}^{\infty} ||x_{t+1} - x_t|| \leq ||L||x_0$ for all possible trajectories $x_1, x_2, ...$ Then following [53] it can be shown that there exists a norm for which $\mathbb{F}$ is paracontractive.

Theorem 3.3.3 approves our observation that if the switching sequences σ and σ' have the same initial subsequence up to a sufficiently large time, i.e. $\sigma[i] = \sigma'[i]$ for $i = 1, ..., N$ where N is large enough then the infinite remainder of sequence in not significant enough and we will have $||x_{ss} - x_{ss}|| < \epsilon$ for any desired ϵ. Furthermore, since $\mathbb{F}$ is a continuous LCP set, infinite products from $\mathbb{F}$ converge with a geometric rate, which is determined in the next section.

3.4 Review of the speed of convergence for consensus problems

A very important measure of performance for a distributed algorithm is its speed of convergence. The convergence speed in a consensus protocol is a functions of the topology of the underlying graph as well as the choice of weights each agent uses to update its value. The problem of determining the speed of convergence for consensus problems is related to the mixing properties of Markov chains and expansion properties of graphs. In fact if we consider a fixed topology, the convergence rate of the consensus protocol is equal to the convergence rate to the stationary distribution of the Markov chain corresponding to the stochastic matrix F. The convergence rate of a Markov chain to its stationary distribution can be determined by algebraic methods as well as geometric and combinatorial methods. The mixing time of a Markov chain is the time required for the

chain to almost forget its initial distribution and converge to its stationary distribution and is defined by:

Definition 3.4.1. *Mixing time*

$$M_\epsilon = min\{T|\forall t \geq T, \pi_0 : ||\pi_0 P^t - \pi||_1 \leq \epsilon\}. \tag{3.13}$$

The algebraic approach to determining the convergence rate of consensus problems and mixing rate of Markov chains uses the celebrated Perron-Frobenius theorem for non-negative matrices:

Theorem 3.4.1. *(Perron-Frobenius) Let F be an aperiodic and irreducible n by n non-negative matrix. The largest eigenvalue of F, λ_1 is single, positive, real and has the largest modulus among all the eigenvalues. The corresponding left and right eigenvectors of F, u_1 and v_1 can be selected to be positive (entry-wise). Denote the rest of eigenvalues of F as $\lambda_2, \lambda_3, ..., \lambda_r$, such that $\lambda_1 = 1 > |\lambda_2| \geq |\lambda_3| \geq ... \geq |\lambda_n|$, and let m_2 be the algebraic multiplicity of λ_2, then*

$$F^t = \lambda_1^t v_1 u_1^T + O(t^{m_2-1}|\lambda_2|^t) \tag{3.14}$$

Here, $O(f(t))$ is a function of t with $\alpha, \beta \in R$, and $0 < \alpha \leq \beta < \infty$, such that $\alpha f(t) \leq O(f(t)) \leq \beta f(t)$ for all t sufficiently large.

Proof. See [101]. □

Therefore, for the consensus iteration with irreducible and aperiodic weight matrix F, $\lambda_1 = 1, u_1 = \pi$, and $v_1 = \mathbf{1}$, and all the other eigenvalues are inside the unit circle:

$$F^t = \mathbf{1}\pi^T + O(t^{m_2-1}|\lambda_2|^t) \tag{3.15}$$

This shows that the consensus algorithm converges with a rate equal to the second largest eigenvalue modulus (SLEM) of F, $\mu(F) = max\{\lambda_2(F), -\lambda_n(F)\}$.

The above analysis is valid if the subdominant eigenvalues of F are not defective [105]. Since such pathologies do not occur in our framework we do not further pursue this matter here.

The spectral gap of a graph, $1 - \mu$ is the difference of the first two largest modulus eigenvalues of the F matrix. Graphs with higher spectral gaps converge more quickly. A direct result of the Perron-Frobenius theory is that

$$||\pi_0 P^t - \pi|| \leq K(n)\mu^t,$$

where $K(n)$ is a constant determined by the number of agents, n.

For the general case, in which topology changes are also included, Blondel et al. ([14]) show that the *joint spectral radius* of a set of matrices derived from the iteration matrices determines the speed of convergence. Joint spectral radius defined as below is a measure of growth of matrix products.

For Σ a set of finite $n \times n$ matrices, their joint spectral spectrum is defined as

$$\rho(\Sigma) = \limsup_{t \to \infty} \max_{A_1, \ldots, A_t \in \Sigma} ||A_t \ldots A_1||^{1/t} \tag{3.16}$$

Consider a set of finite stochastic matrices $\mathbb{F}$ as before. Since these matrices are stochastic they have a spectral radius equal to 1. The product of two stochastic matrices is again stochastic and thus the joint spectral radius of any set of stochastic matrices is equal to 1. To analyze the convergence properties, it is necessary to remove the largest eigenvalue 1 and go to a space of smaller dimension.

To do this, [14] considers a matrix $P \in R^{(n-1)\times n}$ defining a projection on the space orthogonal to $span\{1\}$. So, $P1 = 0$ and if $x^T 1 = 0$ then $||Px||_2 = ||x||_2$. Associated with each $F(t)$ there is a unique matrix $F'(t) \in R^{(n-1)\times(n-1)}$ that satisfies $PF(t) = F'(t)P$. The spectrum of F' is the same as spectrum of F after removing one multiplicity of the eigenvalue 1. If we consider all the matrices F to be connected, then the spectral radius of all matrices F' is less than 1. Let Σ denote the set of all matrices F'. It is known that for any $q > \rho(\Sigma)$, any sequence of matrices $F' \in \sigma$ and any vector y:

$$||F'_1...F'_k|| \leq Cq^k||y|| \tag{3.17}$$

Now, let $\gamma = 1^T x(t)/n$ be the average of elements of the vector $x(t)$, then:

$$Px(t) - P\gamma 1 = Px(t) = PF(t)F(t-1)...F(0)x(0) = F'(t)F'(t-1)...F'(0)x(0) \tag{3.18}$$

Since $(x(t) - \gamma 11)^T 1 = 0$, we have

$$||x(t) - \gamma 1||_2 \leq Cq^t||x(0)||_2 \tag{3.19}$$

for some C and any $q > \rho(\Sigma)$. Assume now that $lim_{t\to\infty} x(t) = c1$ for a constant c. Because all matrices F are stochastic, c must belong to the convex hull of all the entries of $x(t)$ for all t. Therefore:

$||x(t) - c1||_\infty \leq 2||x(t) - \gamma 1||_\infty \leq 2||Px(t) - P\gamma 1||_2$.

and this implies that:

$$||x(t) - c1||_\infty \leq 2Cq^t||x(0)||_2 \tag{3.20}$$

The above result of [14] implies that $\rho(\Sigma)$ gives a measure of the convergence rate of $x(t)$ towards its limit $c1$. In practice, calculation of the joint spectral spectrum of a set of ma-

trices is a mathematically hard problem and is not tractable for sets of large matrices.The ergodic coefficient [101, 18] is defined by:

$$\tau_1(F) = max_{x \in W, ||x||_1 = 1} ||x'F|| \tag{3.21}$$

in which W is the orthogonal subspace to $\mathbf{1}$.

This means that the matrix F contracts the subspace W by at least $\tau_1(F)$ at each iteration. If we denote the ith row of F by f_i, the ergodic coefficient can be written as:

$$\tau_1(F) = \frac{1}{2} max_{i \neq j} ||f_i - f_j|| \tag{3.22}$$

Also for any two stochastic matrices F_1 and F_2, $\tau_1(F_1F_2) \leq \tau_1(F_1)\tau_1(F_2)$. [53]

The ergodic coefficient provides a tractable upper bound for the SLEM and if the F matrices are ergodic, it provides a computable geometric rate to steady state in many cases. For Σ a finite set of $n \times n$ stochastic matrices, $\tau_1(\Sigma)$ is defined as $max_{F \in \Sigma} \tau_1(F)$.

It is worthwhile to notice that graphs with "well-connected" nodes guarantee fast convergence and switching over such graphs will maintain the fast convergence. This is a direct result of Cheeger-type inequalities [1], which relate the spectral gap of the iteration matrices to the geometric properties of the underlying graphs such as expansion . The relationship between the spectrum of a graph and its expansion ratio was first studied by Alon [1]. For a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, the *expansion ratio* (Cheeger constant) is defined as:

$$h(\mathcal{G}) = \min_{S \subseteq \mathcal{V}, |S| \leq \frac{|V|}{2}} \frac{|\partial S|}{|S|}, \tag{3.23}$$

where ∂S denotes the set of edges connecting S to its complement. Denote the maximum degree of nodes in the graph by $\bar{\delta}$. Alon [1] proved that

$$\lambda_{n-1}(L(\mathcal{G})) \geq \frac{1}{2\bar{\delta}} h(\mathcal{G})^2.$$

Desai and Rao [31] explicitly show that the rate of convergence in reversible Markov chains is a function of graph topology as well as the weights on the edges. They introduced a notion of *skewness,* s_π, for such chains:

$$s_\pi = \max_{i,j \in \mathcal{V}} \frac{\pi_i}{\pi_j},$$

and proved that

$$\mu \leq 1 - \alpha \frac{\lambda_{n-1}(L)}{s_\pi},$$

where α is the smallest non-zero off-diagonal entry in F. Therefore, the rate of convergence is a function of graph topology (abstracted out by the Fiedler eigenvalue, $\lambda_{n-1}(L)$), as well as weights (abstracted out by skewness, s_π).

For a given graph topology, the problem of finding optimal weights in the case of doubly stochastic matrices has been addressed in [122]. On the other hand, the effect of graph topology on the convergence of consensus problems and other distributed algorithms has gained lots of interest. Typically, for a collaborative control/ sensing application we are interested in sparse graphs which on the other hand provide high convergence rates and are robust to topology changes. This may seem counter-intuitive and paradoxical. However, classes of graphs such as small world models and expanders are shown to satisfy these contradictory requirements to some degree. We will address the design and small world property in the next chapters.

Expander graphs are $k-$regular graphs on n nodes whose *expansion ratio* is bounded away from zero. The relationship of the expansion ratio of graphs to the spectrum of their adjacency matrices has been studied [1] and it has been shown that random walks mix rapidly on expanders ([56] and references therein). Explicit construction of expander

graphs whose spectral gaps is larger than the asymptotic upper bounds are known as Ramanujan graphs. Accounts of the applications and construction of expanders, Cayley graphs, and Ramanujan graphs in recent control literature can be found in [21], [84], and [63].

3.5 Performance measures for distributed algorithms and their relation to consensus problems

In the previous section we reviewed the results concerning consensus problems and their convergence properties. In general, distributed decision making in networked systems depend critically on timely availability of critical information. The performance of networked systems from the perspective of achieving goals and objectives in an efficient manner is constrained by their collaboration and communication structures.[2] Among the performance measures defined for the distributed algorithms, the most relevant from the perspective of control engineering are the speed of convergence, robustness to losses and failures, and energy/communication efficiency. Since all of these performance measures cannot be achieved at once, we are interested in topologies which favor efficient trade-offs between the level at which these measures can be fulfilled.

Although for any dynamical system or algorithm running on a network, each of these performance measures depend on the particular dynamics of the system, there is an inherent structural effect that makes the performance measures derived for consensus problems applicable to many general distributed systems as well. One reason for this,

[2]For a distributed scenario the collaboration and communication graphs are not necessarily the same.

is the fact that information propagation for distributed algorithms, depends on message passing between the neighbors, and therefore the measures derived for diffusion-type algorithms such as consensus are relevant. Many graph invariants and parameters such as expansion parameters [1] and the number of spanning trees can be determined from the spectrum of the matrices related to graphs, most significantly the Laplacian matrix of the graphs. For example, in many applications, which include message passing among the agents, it is favorable if the graph has a low diameter. On the other hand, the Fiedler eigenvalue of the Laplacian of a graph can be used to provide lower bounds for the diameter and the mean distance in the graph [77]:

$$\text{Diam}(\mathcal{G}) \geq \frac{4}{n\lambda_{n-1}(L(\mathcal{G}))}.$$

The Fiedler eigenvalue also, provides a lower bound on vertex connectivity κ_0 and edge connectivity κ_1 in the graph [77] :

$$\lambda_{n-1}(L(\mathcal{G})) \leq \kappa_0(\mathcal{G}) \leq \kappa_1(\mathcal{G}).$$

In systems of agents with nonlinear dynamics, the effect of topology on the performance of system is also manifested by the eigenvalues of Laplacian matrix of the graph. Slotine et al. [102] show that synchronization in a network of oscillators is critically dependent on the eigenvalues of the Laplacian. They showed that for inefficient topologies such as ring, the threshold of coupling strength for synchronization goes to infinity as the number of agents increase, whereas the same threshold is of order 1 for efficient topologies. Similar results are reported by Jadbabaie et al. [58] for the stability of the Kuramoto model of coupled nonlinear oscillators.

In the sequel, we include an example from majority voting, which emphasizes the importance of being well-connected for majority voting schemes and serves as an inspiration for the hierarchical topology design of chapter 5. We will also consider the robustness of the connectivity graph as a function of the spectrum of the Laplacian matrix, which will be further developed in the next chapter. These two examples emphasize that efficient topologies require favorable trade-offs between performance measures of the system.

3.5.1 Local Majority voting

A classic example by Peleg [92] shows that in voting schemes a few number of well connected nodes can determine the outcome of the process. Consider n citizens each living on a vertex of a graph. Each citizen has an opinion about voting "Yes" or "No" on a controversial subject. However, citizens observe a rule by which they first ask privately their neighbors' opinions. Each person then casts their vote. They will cast "Yes"(resp. "No") if the majority of their neighbors -including themselves- are "Yes-voters"(resp. "No-voters"). An important question is: what is the minimum number of "No-voters" that can guarantee a "No" outcome. As shown in Figure 3.2 the answer is 2. Every one of the $n-2$ "Yes- voters" should change its vote, because of having two "No-voters" in their neighborhood.

It is worthwhile to notice that all the "Yes-voters" observe a 2 to 1 majority of "No-voters" in their neighborhood. However, each "No-voter" observes a huge majority of "Yes-voters" in their neighborhood. Now, consider that the nodes follow the polling rule

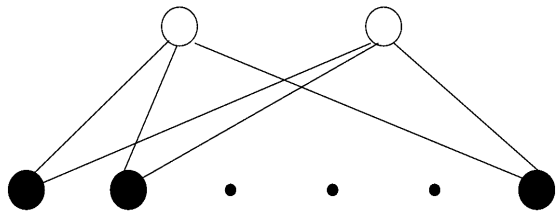


Figure 3.2: Two "No-voters" are enough to control the local majority poll. [92]

iteratively. In this case, each node will oscillate between Yes and No. However, if the "No-Voters" fail to observe the rule, the iteration will converge and all the nodes will vote No after the first iteration.

Peleg [92] also shows that for large n, a negligible minority of $2\sqrt{n}$ "No-voters" can force all the voters to decide to vote No in just one iteration. This can be achieved by a clique of well connected "No-voters" who are attached to groups of badly connected "Yes-voters" as in Figure 3.3. In this case, by following the rules the "No-voters" can force the "Yes-voters" to change their vote while maintaining their own No votes. Therefore the iterative scheme will converge to an "all-No" configuration in just one step.

In this example, the reason that the "No-voters" are dominant is that the connectivity of the subgraph of "No-voters" is superior to the connectivity of the subgraph of "Yes-voters". The connectivity of a subgraph of a graph is characterized by the eigenvalues of such subgraph with appropriate boundary condition. [23]

3.5.2 Robustness to link losses and spanning trees

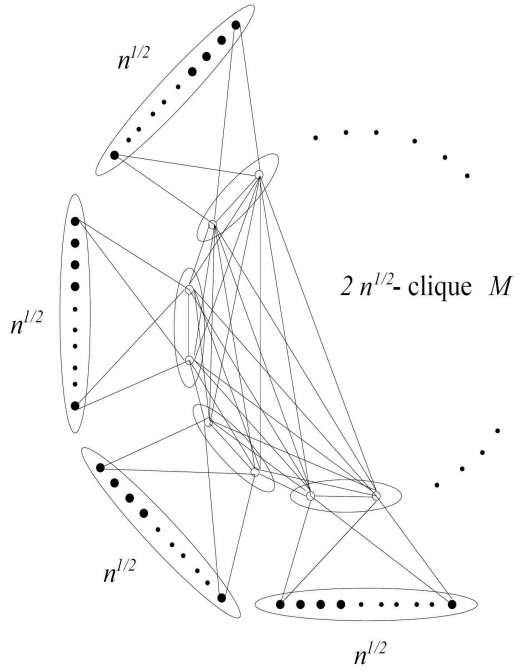


Figure 3.3: A small coalition is enough to control the local majority poll. [92]

The number of spanning trees in a graph is an structural property of it which has many implications for the dynamic systems and algorithms running on it. As indicated in theorems 3.2.4 and 3.2.5, existence of spanning trees in graphs is crucial for the convergence of consensus problems. Intuitively, among all graphs on n nodes, with a given number of edges, the graphs with higher number of spanning trees, are more efficient. Interestingly, in the literature of reliability theory, the number of spanning trees in the graph abstraction of a networked system is considered as a global indicator of network robustness to link losses, when all- to-all connectivity matters. This is the case which we are also interested in: designing graph topologies with good robustness to link losses.

Several measures have been proposed for characterizing the robustness of networks to link losses. Colbourn [25] provides a thorough literature survey on the combinatorics

of network reliability. The classical approach in determining network reliability is to consider constant link loss probabilities for network edges and associate to each network configuration a polynomial, which determines the probability of connectedness of the corresponding configuration. So, the network is modelled as a stochastic graph. We are interested in the case were all of the nodes send information that is intended to be used by all other agents, and therefore it is crucial to have a measure which considers reliable communication between all nodes (all to all). If the link loss probability is high, maximizing the number of spanning trees is essential for robustness of such systems [119, 113]. Kelmans [64, 65, 66] has most prominently studied the problem of graphs with the largest number of spanning trees. Tsen et al. [113] consider an algorithmic approach to finding the most vital edges for the number of spanning trees.

The number of spanning trees of a graph is a metric for well-connectedness of the graph in many applications [119, 25]. In a network with probabilistic link losses, the probability that there exists a path between any pair of nodes is equal to the probability of existence of a spanning tree. In classic reliability theory, a "reliability polynomial" is defined which determines how robust the network is to link losses. Consider a graph $\mathcal{G}$ with n nodes and e edges, a constant probability of link loss p, and let N_i denote the number of connected spanning subgraphs of the graph $\mathcal{G}$. The reliability polynomial [25] is defined as:

$$Rel(\mathcal{G}, p) = \sum_{i=n-1}^{e} N_i (1-p)^i p^{e-i}$$

Denote the number of spanning trees of graph G by $\tau(G)$. It can be verified [119] that

for large p,

$$\tau(\mathcal{G})(1-p)^{n-1}p^{e-n+1} \leq Rel(\mathcal{G},p) \leq \tau(\mathcal{G})(1-p)^{n-1}$$

Graphs with high number of spanning trees, also have applications in network security problems. There, an important problem is to address the concept of trust and how it is established among agents based on previously observed or available evidence [6]. Trust establishment can be considered as a path problem on graphs. An agent i's assessment of trustworthiness of agent j can be calculated using the information contained in any path (relational or logical) from agent i to agent j. This problem has been addressed using a semiring method by Theodorakopoulos and Baras [112]. In such methods any spanning tree of the graph corresponds to a minimal graph which is necessary for all-to-all trust establishment. A larger number of spanning trees corresponds to richer basis for trust establishment. Reference [72] uses a probabilistic models from reliability methods for trust assessments, in which the spanning trees are crucial.

The number of spanning trees in a graph can be determined from the Laplacian of the graph as well as the consensus type natural random walk matrix corresponding to the graph. We use a similar graph theoretic abstraction to represent the network as in the consensus problem. We consider a set of n agents and model the interconnection between them by a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. The nodes of the graph, $\mathcal{V} = \{1, 2, ..., n\}$ represent the agents and the undirected edges $\mathcal{E} = \{l_1, l_2, ..., l_e\}$ represent the links.Given an arbitrary orientation of the edges of graph $\mathcal{G}$, the incidence matrix E of the graph is an n by e matrix, which has $1, -1$ or 0 in the ij^{th} position if the edge j is correspondingly an incoming edge to node i, an outgoing edge from node i, or not incident to node i. The

degree of the i^{th} node, d_i equals the total number of edges incident to it.

The Laplacian of a graph can be written as:

$$L = D - A = EE^T.$$

The normalized Laplacian of a graph, $\mathcal{L}$ is defined as:

$$\mathcal{L} = D^{-1/2}LD^{-1/2}.$$

The normalized Laplacian is closely related to the stochastic transition matrix of the natural random walk on a graph,

$$P = D^{-1}A = I - D^{-1}L.$$

Using the similarity transformation, $D^{1/2}PD^{-1/2}$, it can be verified that

$$\lambda_i(P) = 1 - \lambda_{n+1-i}(\mathcal{L}),$$

where $\lambda_i(.)$ denotes the i^{th} largest eigenvalue.

3.5.2.1 Matrix-tree theorem and its variants

The number of spanning trees in a graph can be determined by Kirchhof's *matrix-tree theorem* [40]. Since for connected graphs L is a positive semi-definite matrix with $\lambda_{n-1}(L) > 0$, the nullspace of L is spanned by $\mathbf{1}$. On the other hand

$$L.Adj(L) = det(L).I_n = 0_n,$$

and L is symmetric; therefore $Adj(L)$ is a constant multiple of $J = \mathbf{1}\mathbf{1}^T$. This constant is equal to the number of the spanning trees of the graph as indicated by the Matrix-tree theorem [40].

Theorem 3.5.1. *Let $\tau(\mathcal{G})$ denote the number of spanning trees in a graph G, $L, \mathcal{L}, P$ denote the Laplacian, normalized Laplacian, and natural random walk matrices of G, $Q = I - P$, and Q_i denote the i^{th} principal sub-matrix of Q, i.e. the matrix obtained by deleting the i^{th} row and column of Q, then:*

1.

$$Adj(L) = \tau(\mathcal{G})L. \tag{3.24}$$

2.

$$\tau(\mathcal{G}) = \frac{1}{n}\prod_{j=1}^{n-1}\lambda_j(L). \tag{3.25}$$

3.

$$\tau(\mathcal{G}) = \frac{1}{n}det(L + \frac{1}{n}J) \tag{3.26}$$

4.

$$\tau(\mathcal{G}) = \prod_{j=1}^{n-1}\lambda_j(\mathcal{L})\frac{\prod_{i=1}^{n} d_i}{\sum_{i=1}^{n} d_i} \tag{3.27}$$

5. If $\mathcal{G}$ is connected,

$$\begin{aligned}\tau(\mathcal{G}) &= \prod_{j=2}^{n}(1-\lambda_j(P))\frac{\prod_{i=1}^{n} d_i}{\sum_{i=1}^{n} d_i} = \sum_{j=1}^{n} det(Q_j)\frac{\prod_{i=1}^{n} d_i}{\sum_{i=1}^{n} d_i} \\ &= det(Q_k)\prod_{i\neq k} d_i, \quad \forall k = 1, ..., n.\end{aligned} \tag{3.28}$$

Proof. (Sketch)

1. The proof of this is classic and can be found in e.g. [40], Lemma 13.2.3. The main idea is to express any cofactor of L as a matrix determined from E by removing a column, using the Binet-Cauchy formula for determinants, and observing that the

contribution of each spanning tree to the sum is 1 and contribution of any subgraph including loops to the sum is zero.

2. This is also classic and follows directly from 1, by equating the coefficient the linear term in the characteristic equation of L and the sum of $n-1$ term products of eigenvalues of L.

3. This follows from 2, since $L + \frac{1}{n}J$ has the same set of eigenvalues as L, except for $\lambda = 0$, which is replaced by $\lambda = 1$. This can be verified easily, since for the eigenvector x_i corresponding to λ_i, $i = 1, 2, ..., n-1$:

$$(L + \frac{1}{n}J)x_i = Lx_i + \frac{1}{n}\mathbf{1}\mathbf{1}^T x_i = \lambda_i x_i,$$

$$(L + \frac{1}{n}J)\mathbf{1} = L\mathbf{1} + \frac{1}{n}\mathbf{1}(\mathbf{1}^T\mathbf{1}) = \frac{1}{n}.n\mathbf{1} = \mathbf{1}.$$

4. The proof is similar to 2, and can be found in [23].

5. This is a direct result of 4 and some properties of the eigenvalues of $I - P$ [75]. Since G is connected, P is irreducible. The eigenvalues of $I - P$ and $\mathcal{L}$ are the same, since:

$$D^{1/2}(I - P)D^{-1/2} = D^{1/2}(D^{-1}L)D^{-1/2}$$
$$= D^{-1/2}LD^{-1/2} = \mathcal{L}.$$

If the characteristic equation of $I - P$ is

$$p(\lambda) = \lambda^n + \alpha_{n-1}\lambda^{n-1} + ... + \alpha_1\lambda + \alpha_0,$$

then similar to part 2, $\alpha_1 = (-1)^{n-1} \prod_{i=2}^{n} \lambda_i = (-1)^{n-1} \sum_{i=1}^{n} det(Q_i)$, and therefore

$$\tau(G) = \sum_{j=1}^{n} det(Q_j) \frac{\prod_{i=1}^{n} d_i}{\sum_{i=1}^{n} d_i}.$$

On the other hand, since the graph is connected the nullspace of $I - P$ is spanned by $\mathbf{1}$. Furthermore,

$$Adj(I - P)(I - P) = (I - P)Adj(I - P) = 0,$$

Therefore the invariant distribution for P can be represented as:

$$\pi^T = \frac{1}{\sum_{j=1}^{n} det(Q_j)} [det(Q_1) \quad det(Q_2) \quad ... \quad det(Q_n)].$$

But since P is the matrix corresponding to the natural random walk on the graph G, $\pi_k = \frac{d_k}{\sum_{i=1}^{n} d_i}$. Hence equation (3.28) results.

□

As we will show in section 4.8, an interesting concept which appears in the problem of optimizing the number of spanning tress, is the effective resistance between the nodes of the graph. Let $Z = (L + \frac{1}{n}J)^{-1}$. Consider an edge $l = (i, j)$. Since l is between nodes i and j, its incidence vector can be written as $f = e_i - e_j$, where e_i denotes a unit vector with 1 in the i^{th} entry. Therefore, we have

$$f^T Z f = z_{ii} - 2z_{ij} + z_{jj}.$$

This quantity is referred to as the *effective resistance* or the *resistance distance* between nodes i and j of the undirected graph $\mathcal{G}$ [39, 4]. If we consider the graph as a resistor network with 1Ω resistors on edges, this is the effective resistance when a voltage difference of $1V$ is applied across edge l.

Chapter 4

Consensus problems on Small world networks and design of efficient topologies

4.1 Basic setup for the analysis and design problems

The main focus of chapters 4 and 5 is to characterize small world networks as "efficient" topologies and to provide guidelines for design of efficient topologies for decision making and control purposes. Consensus problems play an important role in our approach to the problem. Among the different performance measures that one can define for distributed algorithms, we study the speed of convergence and robustness to failures in the presence of connectivity constraints. The connectivity constraints are means for abstracting out the distributed structure of the algorithms and the constraints on communication and processing capabilities of the nodes. We consider a topology to be efficient if it satisfies a certain degree of robustness or fast convergence, given a limit on the number of the links in the network.

To address the problem of small world networks as efficient graphs, we develop a method for investigating the effects of small world topologies by building on the probabilistic models of Higham [55], that establish an equivalent representation of small world topologies as rare (i.e. with small probability) transitions among non-neighboring states in the Markov chain associated with a graph. In the case where the performance mea-

sure is the speed of convergence, we address the problem of characterizing small world networks by studying the improvement in convergence speed of consensus problems on perturbations of nominal base graphs and relate the perturbations of Markov chains associated with graphs to probabilistic switching topologies and shortcuts.

This interpretation of small world graphs and associated phenomena promotes a probabilistic viewpoint towards understanding and quantifying the small world effect. Thus, one can allow time-varying topologies, in which every node nominally communicates according to a pre-defined topology, corresponding to the original graph from which a small world network is obtained. The probabilistic interpretation of the small world topology via perturbations of the associated Markov chain, is that with a small probability the node communicates with non-adjacent nodes at every time step. Since the small world model is obtained by stochastically adding or rewiring a few edges to a nominal graph, we anticipate that adding a small number of long distance edges is analogous to choosing graphs with low probability shortcuts. We propose a model, which is a formalization of this idea. The model may also be considered as switching between multiple topologies to increase the convergence rate. In such an interpretation, communication with remote nodes is done with a very small probability to conserve the node power. To this end we develop a framework for studying consensus problems with probabilistic switching between topologies and address the mean square convergence of consensus problems. The model can be independently used to determine whether for a given topology, probabilistic switching increases the speed of convergence or not.

We propose the following procedure for characterizing the small world effect:

Procedure 1. *1. To a nominal base graph, $\mathcal{G}_0$ correspond a natural random walk matrix, F_0.*

2. *Capture the desired performance measure as a (usually spectral) property of this matrix.*

3. *Take a probability distribution corresponding to a perturbation of the base graph, G_ϵ. In other words, perturb the graph in such a way that the random walk on graph $\mathcal{G}_0$ is modified to be able to make transitions to non-neighboring nodes, with small probability according to a perturbed matrix F_ϵ.*

4. *Determine if for a small perturbation parameter, there is any abrupt increase in the performance measure.*

5. *If such increase can be observed for a range of perturbation parameter, which corresponds to a small perturbation of the base graph, then the small world phenomenon has happened.*

6. *Interpret the perturbations in the weights of the random walk as structural perturbations.*

In the following sections, we derive such procedure as a result of our investigation of the convergence speed of consensus problems on small world networks, and show how the probabilistic and perturbation interpretations are related. The idea is then extended to the robustness measure of number of spanning trees.

The other important problem to investigate is how adding a shortcut or a small set of shortcuts affects the efficiency of a base graph. This can be formalized as follows. Given

an initial graph topology, we are interested to determine how k edges can be added so that the resulting graph topology has the optimal performance measure among all possible topologies. Examples of optimal properties are maximal number of spanning trees and maximal spectral gap for the the consensus problem.

Consider a dynamic graph which evolves in time from a given topology $\mathcal{G}_0 = (\mathcal{V}_0, \mathcal{E}_0)$. Let's denote the complete graph on n vertices by $\mathcal{K}_n$. Also, denote the complement of a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ -which is the graph with the same vertex set but whose edge set consists of the edges not present in $\mathcal{G}$ - by $\bar{\mathcal{G}}$. So, $\mathcal{E}(\bar{\mathcal{G}}) = E(\mathcal{K}_n) \setminus \mathcal{E}(\mathcal{G})$.

If we denote the operation of adding edge e to graph G by $Add(G, e)$, we consider the dynamic graph evolution:

$$\begin{aligned} &G(t+1) = Add(G(t), u(t)), \quad t = 0, 1, ...k-1 \\ &u(t) = e(t+1), \qquad\qquad\qquad e(t+1) \in S \subseteq E(\bar{G}(t)) \\ &G(t) = \mathcal{G}_0 \end{aligned} \tag{4.1}$$

The problem is to:

$$\begin{aligned} \text{maximize} \quad & \mathcal{J}(G(t+k)) \\ \text{subject to:} \quad & (4.1) \end{aligned} \tag{4.2}$$

where $\mathcal{J}(G(t))$ is the performance measure of the dynamic graph and can be equal to $\tau(G(t)) = \prod_{i=1}^{n-1} \lambda_i(L(G(t)))$ or $SLEM(F(G(t))$.

This is a combinatorial optimization problem. If we denote the number of edges of $\mathcal{G}_0$ by e_0, there are $2^{\binom{n}{2}-e_0}$ possible edges, among which we should choose k. Even if we take a smaller candidate edge set S, the search space is very large and exhaustive search is not practical even for moderate graph sizes. An instrumental method to make

such combinatorial problem tractable is to consider a convex relaxation of the problem.

For the case, where the performance measure is the SLEM of the random walk matrix F, it is important to notice that $SLEM(F)$ is not a convex function of F in the general case. The convexity holds if F is symmetric, because for symmetric F we can write:

$$\lambda_2(F) = Sup\{x^T F x | x^T x \leq 1, 1^T x = 0\} \tag{4.3}$$

$$-\lambda_n(F) = Sup\{-x^T F x | x^T x \leq 1, 1^T x = 0\} \tag{4.4}$$

Since $x^T F x$ is an linear function of F and thus convex and Sup preserves convexity, both λ_2 and $-\lambda_n$ are convex. So $SLEM = \max\{\lambda_2, -\lambda_n\}$ is also convex. This does not hold for the general case. A counter intuitive result is that the SLEM does not monotonically change with addition of edges. Consider a $2k-$regular ring structured base graph on n nodes in which each node i is connected to nodes j for which $j = (i + k') \mod n$ or $j = (i - k') \mod n$ for $k' = 0, 1, ..., k$. If n is even, adding an edge will increase SLEM unless a vertex is connected to the farthest vertex from it that is i is connected to $i + n/2$ (modulo 2). In this case one of the multiplicities of the SLEM decreases but the other multiplicity is not changed. Figures 4.1 and 4.2 illustrate this effect. The dotted line tangent to the curves show the SLEM of the original curves. The more distant the two joined vertices, the less increase in SLEM.

Ghosh and Boyd [38] address the optimization of the second smallest eigenvalue (the Fiedler eigenvalue) of the graph Laplacian as a measure of well-connectedness of the graph. The Fielder eigenvalue is a concave function of the matrix L. They relaxed the combinatorial problem to a convex problem, used semidefinite programming to solve it,

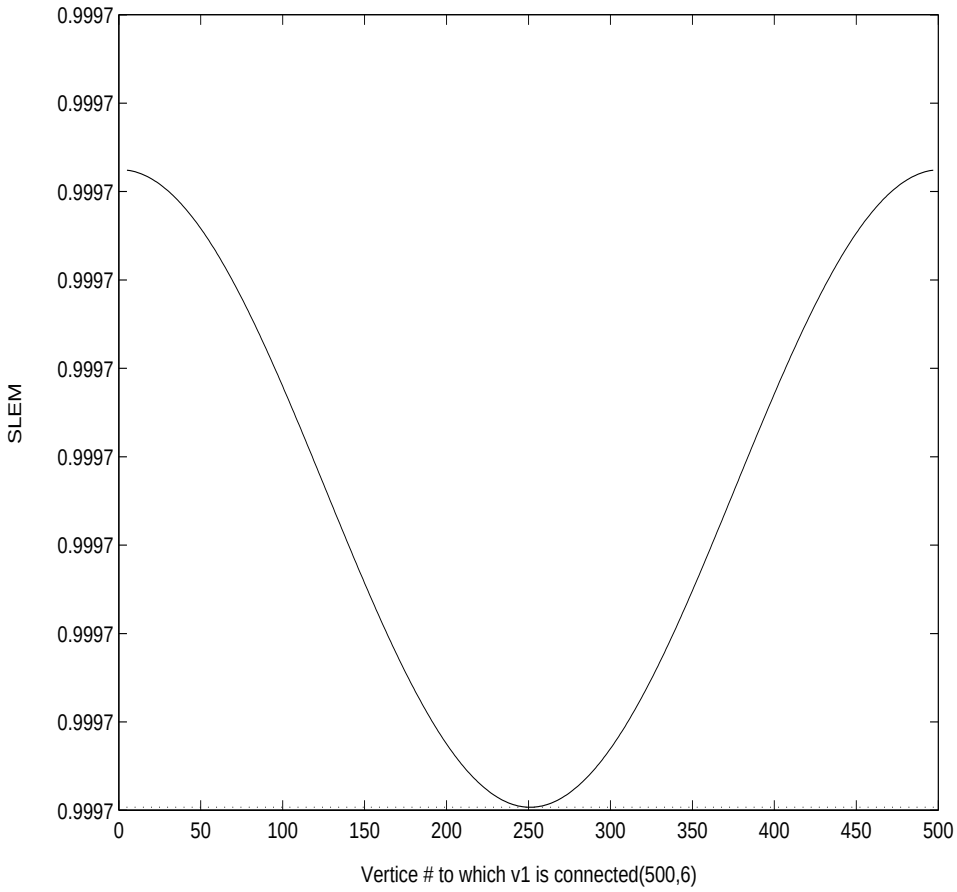


Figure 4.1: Adding a vertex (500,3), The green (dotted) line tangent to curve shows SLEM before adding edge

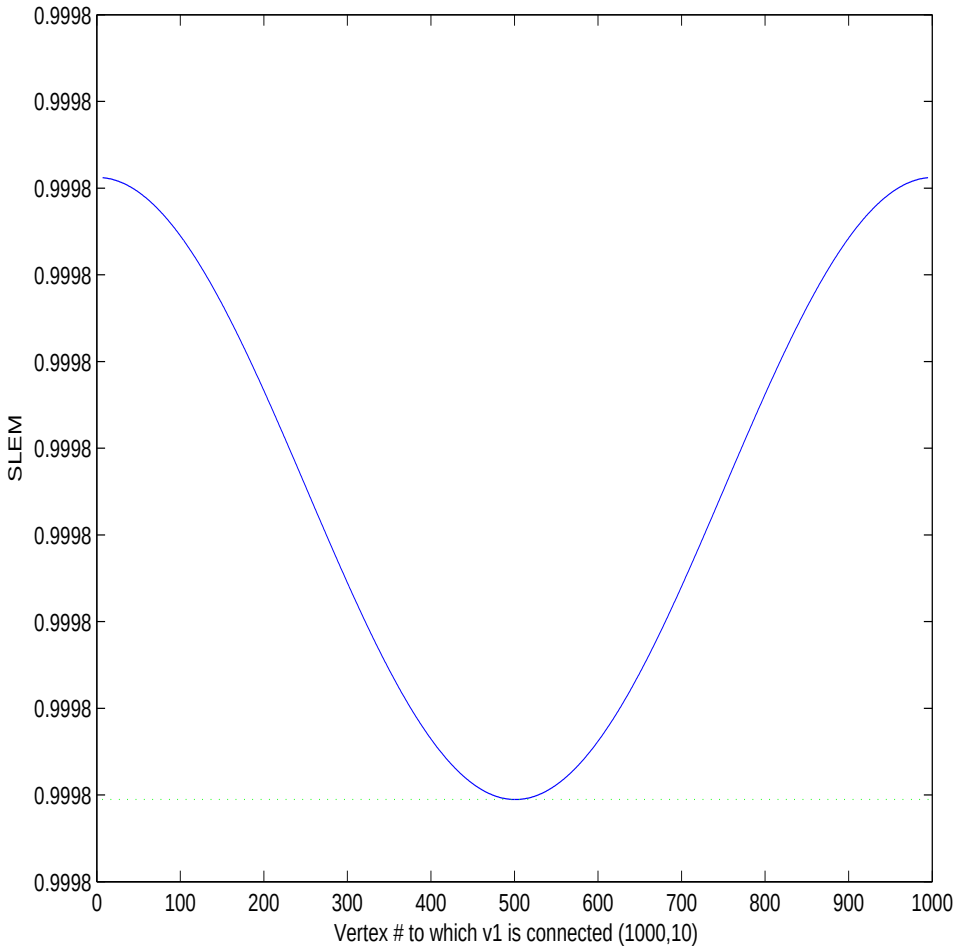


Figure 4.2: Adding a vertex (1000,5), The green (dotted) line tangent to curve shows SLEM before adding edge

and provided a heuristic for large scale graphs. In sections 4.8 and 4.9, we will consider the relaxation for the case with the number of spanning trees in the graph $\tau(\mathcal{G})$ as the performance measure and derive heuristics that provide approximately optimal solutions.

4.2 Consensus problems on small world networks

We study a model of small world networks proposed by Watts and Strogatz [118]. They introduced and studied a simple tunable model for explaining the behavior of many real world complex networks. Their aim was to provide a model to explain two seemingly contradictory characteristics of many networks: existence of short paths between arbitrary nodes and the higher likelihood of adjacency for two nodes which share common neighbors. Their small world model takes a regular ring-structured lattice graph and replaces the original edges by random ones with some probability $0 \leq \phi \leq 1$. It is conjectured that dynamical systems coupled in this way would also display enhanced signal propagation and global coordination, compared to regular lattices of the same size. The intuition is that the short (i.e. direct) paths between distant parts of the network cause high speed spreading of information which may result in fast global coordination. The small world effect in this respect can thus be interpreted as a significant improvement of a performance function with small perturbation to the graph topology.

In the context of consensus algorithms, this conjecture has been tested and shown to be true. Olfati-Saber [82] studied continuous time consensus protocols on small world networks and proposed some conjectures. We consider the discrete time consensus problem on undirected graphs with Vicsek weights -consensus iteration matrices are of the

form of equation (3.6)- i.e. they correspond to natural random walk with a self loop on the graph. Denote the neighbor set of node i at time t by $\mathcal{N}_i(t)$ and the number of neighbors of node i at time t by $n_i(t)$. At every time step t, each node i updates its value according to the equation

$$x_i(t+1) = \sum_{j \in \mathcal{N}_i(t) \cup i} w_{ij}(t) x_j(t) \tag{4.5}$$

where the weight assigned by the node i to the value of node j is given by $w_{ij}(t) = \frac{1}{n_i(t)+1}$ and $\sum_{j \in \mathcal{N}_i(t) \cup i} w_{ij} = 1, \quad \forall i$. Denote the interconnection graph at time t by $\mathcal{G}(t)$. We can associate a matrix $F(t) = (I + D(t))^{-1}(I + A(t))$ with this graph where

$$F_{ij}(t) = \begin{cases} \frac{1}{n_i(t)+1} & j \in \mathcal{N}_i(k) \cup i \\ 0 & \text{otherwise.} \end{cases} \tag{4.6}$$

We consider the Newman-Moore-Watts [81] improved form of $\phi-$model originally proposed by Watts and Strogatz. The model consists of a $2k-$regular ring structured base graph on n nodes in which each node i is connected to nodes j for which $j = (i + k') \mod n$ or $j = (i - k') \mod n$ for $k' = 0, 1, ..., k$. Shortcut links are added -rather than rewired- between randomly selected pairs of nodes, with probability ϕ per link on the underlying lattice; thus there are typically $nk\phi$ shortcuts. Here we actually force the number of shortcuts to be equal to $nk\phi$(comparable to Watts ϕ model.) We consider the relative increase in the convergence rate (compared to the base lattice) as the performance function, denote it by Δ and call it spectral gap gain:

$$\mathcal{J}(\mathcal{G}_\phi) = \Delta = \frac{1 - \mu(F_\phi)}{1 - \mu(F_0)} \tag{4.7}$$

In the above equation F_ϕ denotes the consensus iteration matrix corresponding to the

perturbed graph, whereas F_0 denotes the consensus iteration matrix corresponding to the base lattice.

We consider different initial topologies $(n, k) = (100, 2), (200, 3), (500, 3), (1000, 5)$, generated 20 samples of small world graphs $\mathcal{G}_\phi$ for different ϕ values chosen in a logarithmic scale between 0.01 and 1. These choices of (n, k) are selected so that we can compare are results with those of [82]. In the figures 4.3 and 4.4, we have depicted the gain in spectral gap of the resulted small world graphs with respect to the spectral gap of the base lattice. Only the results of cases $(500, 3)$ and $(1000, 3)$ are demonstrated since the other topologies follow a similar pattern. We observe that:

1. In the low range of ϕ ($0 < \phi < 0.01$) no spectral gain is observed and the SLEM is almost constant.

2. A phase transition in the spectral gap curves is observed around $\phi = 0.1$.

3. Simulations show that small world graphs posses good convergence properties as far as consensus protocols are concerned.

Analytical verification of the small world effect in consensus problems can be addressed by both random graph methods and perturbation methods. The random graph approach considers an ensemble of graphs with links which are present with small probability and determines high probability results. Durrett [34] studied mixing times of random walks on different types of small world and random graphs. The spectral properties of such random graphs can be related to their diameter using the Cheeger's inequality. This approach has been used by [107] to find high probability results for consensus problems on small world graphs. Reference [35] and [91] used similar methods.

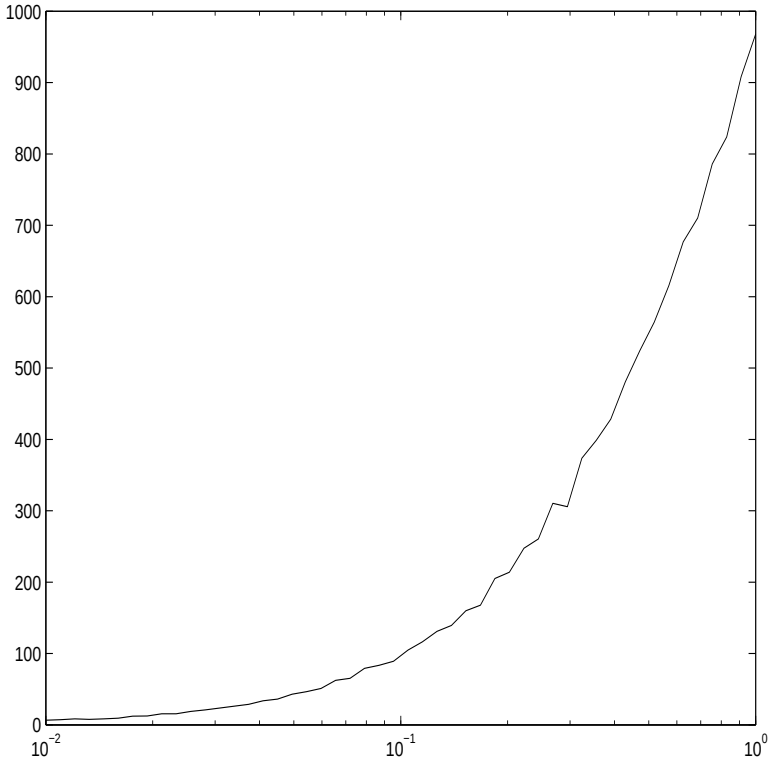


Figure 4.3: Spectral gap gain for (n,k)=(500,3)

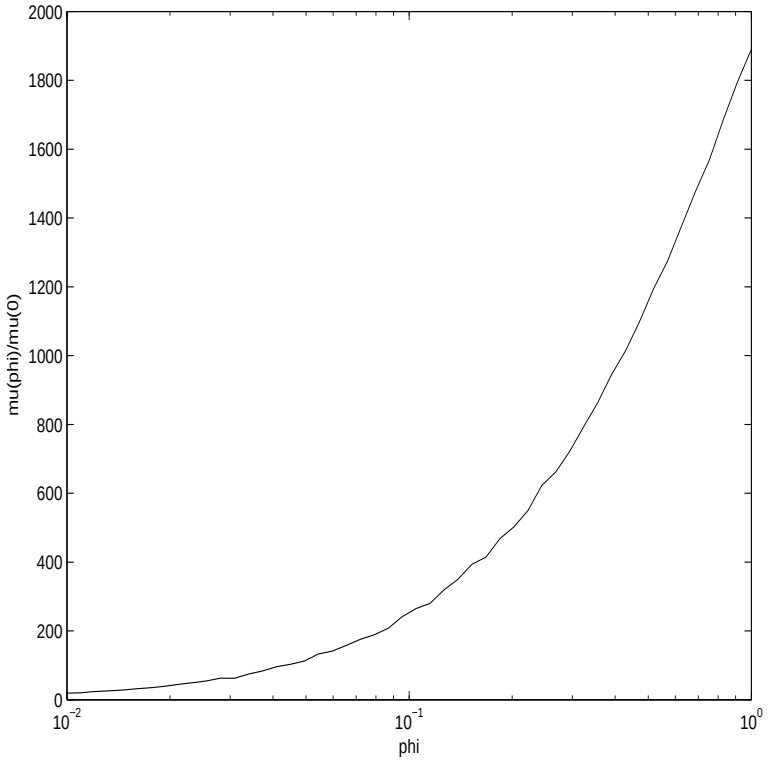


Figure 4.4: Spectral gap gain for (n,k)=(1000,5)

Another method to study the small world networks is by using perturbation methods. To best of our knowledge, this was first proposed by Higham in [55], who considered the hitting time of an arbitrary node in random walk on a ring based small world graph. Following Higham, we use perturbation and a "mean field " method to justify our results. We generalize this result by considering the ability of an arbitrary base graph to become a small world.

4.3 Perturbation Analysis

We consider the variant of the ϕ model of Small World graphs mentioned in section 4.2, in which a regular lattice is considered and m shortcuts are added randomly, where m is equal to a proportion ϕ of the lattice's initial edges. In the present analysis, we reflect the effect of shortcuts by adding "small" nonzero positive numbers to the entries of F corresponding to non-adjacent nodes of the lattice. The small perturbation corresponds to using lots of shortcuts with negligible weights on them. Following [55], we call such perturbations $\epsilon-$shortcuts. Although by adding a uniform perturbation the topology of the graph is not respected, the analysis gives insight on random communication patterns for Small World networks. The results may be reinterpreted in a probabilistic framework where at each time instant the topology of the graph is respected.

The model starts with a ring-shaped lattice of n nodes, each connected by undirected nodes to its k nearest with $k \ll n$. We call this the base graph and denote it by $\mathcal{G} = C(n, k)$ and the corresponding F matrix by F_0:

$$F_0 = \begin{pmatrix} a_1 & a_2 & a_3 & . & . & . & a_n \\ a_n & a_1 & a_2 & . & . & . & a_{n-1} \\ a_{n-1} & a_n & a_1 & . & . & . & a_{n-2} \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ a_2 & a_3 & . & . & . & . & a_1 \end{pmatrix} = circ[a_1, a_2, ...a_n] \tag{4.8}$$

in which:

$$a \triangleq [a_1, a_2, ...a_n] = \tfrac{1}{2k+1}[\underbrace{1, ..., 1}_{k+1} \quad \underbrace{0, ..., 0}_{n-2k-1}, \quad \underbrace{1, ..., 1}_{k}] \tag{4.9}$$

The following lemma holds.

Lemma 4.3.1. *The second largest eigenvalue modulus (SLEM) of F_0 has multiplicity at least 2. Furthermore as $n \to \infty$,*

$$1 - \mu(F_0) = O(n^{-2}). \tag{4.10}$$

Proof. Let $\omega = e^{\frac{2\pi\sqrt{-1}}{n}}$ be the nth root of unity. The matrix F_0 is circulant and $F_0 = circ(a)$, where a is as in (4.9). The representer [30] of this circulant is

$$p_a(z) = \frac{1}{2k+1}(1 + z + ... + z^{k-1} + z^k + z^{n-k} + z^{n-k+1} + ... + z^{n-1}) \tag{4.11}$$

So, the eigenvalues of this matrix are $\lambda_i = p_a(\omega^{i-1})$. Since F_0 is stochastic, $\lambda_1 = 1$ and moreover it is a simple eigenvalue because the underlying graph is connected. Since for

integers A and B, $\omega^{An+B} = \omega^B$, it follows that $\lambda_2 = \lambda_n$, $\lambda_3 = \lambda_{n-1}$ and so on. In the case that n is odd apart from $\lambda_1 = 1$, all eigenvalues come in pairs. In the case that n is even, $\lambda_{\frac{n}{2}+1}$ is the only eigenvalue which can be simple, however direct calculation shows that it is equal to $\frac{(-1)^k}{2k+1}$ which is clearly less than $\lambda_2 = \lambda_n$. Since $k \ll n$, $\mu = \lambda_2 = \lambda_n = \frac{1}{2k+1}[1 + 2Re(\omega) + 2Re(2\omega) + ... + 2Re(k\omega)] < 1$ and $\lambda_i \leq \lambda_2$ for $i \in 2, ..., n-1$. Using trigonometric identities we get:

$$\lambda_2 = \frac{1}{2k+1}\left[1 + 2\frac{\frac{\cos(k+1)\pi}{n}\sin\frac{k\pi}{n}}{\sin\frac{\pi}{n}}\right]$$

Using the Taylor expansion of the involving functions yields:

$$1 - \mu(F_0) = \frac{\alpha n^{-2} + O(n^{-3})}{(2k+1)\pi + O(n^{-1})} = \beta n^{-2},$$

where α and β are nonnegative constants. □

Now we perturb the nonzero entries of the matrix F_0 by $\epsilon = \frac{K}{n^\alpha}$ for fixed $K > 0$ and $\alpha \geq 1$. These are the $\epsilon-$shortcuts. To preserve the stochasticity of the matrix, we also change the nonzero elements of F_0 in a uniform manner. The resulting matrix F_ϵ has the following structure:

$$(F_\epsilon)_{ij} = \begin{cases} \frac{1}{2k+1} - \frac{n-2k-1}{2k+1}\epsilon & \text{If } (F_0)_{ij} = \frac{1}{2k+1} \\ \epsilon & \text{otherwise.} \end{cases} \tag{4.12}$$

The following statement presents the effect of $\epsilon-$shortcuts on the convergence rate, and shows that the small world effect happens as a result of the continuous increase of ϵ.

Proposition 4.3.2. *Let* $\epsilon = \frac{K}{n^\alpha}$, $\alpha \geq 1$.

- *For $\alpha > 3$, the effect of ϵ-shortcuts on convergence rate is negligible. $\alpha = 3$ is the onset of the effectiveness of shortcuts.*

- *For $\alpha = 2$, the shortcuts are dominantly decreasing SLEM.*

- *For $1 < \alpha < 2$, almost all of the nodes communicate effectively and thus the SLEM is very small.*

Proof. The matrix F_ϵ is also circulant and its eigenvalues can be computed. The representer [30] of this circulant is

$$p_a(z) = \left[\frac{1-n\epsilon}{2k+1} + \epsilon\right](1+z+...+z^{k-1}+z^k+z^{n-k}+z^{n-k+1}+...+z^{n-1})+\epsilon(z^{k+1}+...+z^{n-k-1}) \tag{4.13}$$

Using some algebra and trigonometric identities yields that for $k \ll n$,

$$\mu(F_\epsilon) = \frac{1-n\epsilon}{2k+1}\left[1 + 2\frac{\frac{\cos(k+1)\pi}{n}\sin\frac{k\pi}{n}}{\sin\frac{\pi}{n}}\right] \tag{4.14}$$

The spectral gap can be approximated by:

$$1 - \mu(F_\epsilon) \approx \frac{n\epsilon}{2k+1} - \frac{2kn\epsilon(k+1)^2\pi^2}{(2k+1)(2n^2)} + h.o.t.$$

Substituting $\epsilon = Kn^{-\alpha}$ and comparing this to the spectral gap of the base lattice yields the desired results. □

For the base lattice, the spectral gap decreases as fast as n^2. If ϵ is $O(n^\alpha), \alpha > 3$, then terms coming from the lattice are dominant, and therefore the shortcuts does not affect the spectral gap. For $\alpha = 3$ the terms regarding the shortcuts will be of the same degree as the terms from the base and for K large enough, the SLEM starts decreasing from the corresponding lattice SLEM. For $\alpha = 2$ the terms regarding the shortcuts are

dominant and the SLEM has considerably decreased compared to the base lattice. Decreasing α further will cause the spectral gap not vanish as $n \to \infty$. In fact using $\epsilon = \frac{1}{n}$ corresponds to the fully centralized scheme, in which all the weights are equal. In this case $\mu(f_\epsilon) = 0$ and the spectral gap reaches its maximum.

As observed above $\epsilon-$shortcuts are loosely analogous to the shortcuts in the ϕ-model in the sense that if we consider a random walk on the graph, using $\epsilon-$shortcuts corresponds to the random walk using a shortcut, which takes place with much lower probability than other links. Since the Small World model is a probabilistic model, we anticipate that adding small weights is analogous to choosing graphs with low probability shortcuts. This idea is developed in section 4.6.

4.4 Stochastic characterization of the small world effect

In the previous section, we considered the small world model of Watts and Strogatz. In this section, we consider a general setting in which the base graph can be any graph. However, usually it is considered to be sparse due to obvious practical reasons having to do with notions of efficiency. We are interested in the question:

1 Is it possible to increase the convergence rate of the consensus algorithm on a given [sparse] graph substantially by adding a set of few random shortcuts?

Here we will again address a relaxed "mean-field" analysis of the model, i.e. we consider the question:

$1'$ Is it possible to increase the convergence rate of the consensus problem on a given [sparse] graph substantially by adding small $\epsilon-$shortcuts?

In [33], Duchon et al. use the term "small-worldizable" for a set of graphs which can be turned into small world in the sense of finding poly-logarithmic paths between arbitrary nodes. We use their terminology "small-worldizable" in our context. We give a characterization of the small world effect for any general base. The following procedure gives an implicit definition of what we mean by being "small-worldizable. As before we denote the second largest eigenvalue modulus (SLEM) of the matrix F by $\mu(F)$.

Definition 1. Small-worldizable graphs

Given a graph $\mathcal{G}_n$ on n vertices:

- *Consider a natural random walk on this graph. Denote the corresponding Markov Chain matrix as*

$$F_0 = (I + D)^{-1}(A + I)$$

where A is the adjacency matrix of the graph $\mathcal{G}_n$ and D is the diagonal matrix with each node's degree on the corresponding diagonal.

- *Perturb the zero elements of F_0 by a small positive $\epsilon = \epsilon(n)$, such that $\lim_{n\to\infty} \frac{\epsilon(n)}{n} = 0$ and adjust its nonzero elements so that the resulting matrix F_ϵ remains stochastic.*

- *A graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ is considered to be small-worldizable if $\frac{\Delta(F_\epsilon)}{\Delta(F_0)} \gg 1$, where $\Delta(F)$ denotes the spectral gap $1 - \mu(F)$.*

- *For a small-worldizable graph, we refer to the value range of ϵ for which $\frac{\Delta(F_\epsilon)}{\Delta(F_0)} = O(1)$, the* Onset of the small world effect.

Consider a sparse connected graph $\mathcal{G}$, and its corresponding Random walk matrix F_0. We perturb F to get F_ϵ

$$(F_\epsilon)_{ij} = \begin{cases} \epsilon & (F_0)_{ij} = 0 \\ (1 - n\epsilon)(F_0)_{ij} + \epsilon & (F_0)_{ij} \neq 0 \end{cases}$$

where $\epsilon < 1/n$. Therefore we can write:

$$F_\epsilon = (1 - n\epsilon)F_0 + \epsilon \mathbf{1}\mathbf{1}^T.$$

Then we obtain the result:

Theorem 4.4.1. *The graph $\mathcal{G}$ is small-worldizable if and only if $\frac{\mu}{1-\mu} \gg \frac{1}{n\epsilon}$.*

The proof of Theorem 4.4.1 is based on the following Lemma:

Lemma 4.4.2. *The second largest eigenvalue modulus (SLEM) of F_ϵ is given by*

$$\mu(F_\epsilon) = (1 - n\epsilon)\mu(F_0)$$

.

Proof. Consider the matrix

$$F_1 = (1 - n\epsilon)^{-1} F_\epsilon = F_0 + \frac{\epsilon}{1 - n\epsilon}\mathbf{1}\mathbf{1}^T.$$

From the Sherman-Morrison-Woodbury formula we have

$$\det(F_1 - \lambda I) = \det(F_0 - \lambda I + \frac{\epsilon}{1 - n\epsilon}\mathbf{1}\mathbf{1}^T)$$

$$= [1 + \frac{\epsilon}{1 - n\epsilon}\mathbf{1}^T(F_0 - \lambda I)^{-1}\mathbf{1}] \det(F_0 - \lambda I) \tag{4.15}$$

Furthermore, for any $\lambda \notin Spec(F_0)$,

$$(F_0 - \lambda I)^{-1}\mathbf{1} = (1 - \lambda)^{-1}\mathbf{1}$$

$$\det(F_1 - \lambda I) = \det(F_0 - \lambda I + \frac{\epsilon}{1 - n\epsilon}\mathbf{1}\mathbf{1}^T) \tag{4.16}$$

It follows that the eigenvalues of F_1 are the same as the eigenvalues of F_0 except for $\lambda_1(F_1) = 1 + \frac{n\epsilon}{1-\epsilon}$. Therefore:

$$\lambda_1(F_\epsilon) = 1,$$

and for $i \neq 1$

$$\lambda_i(F_\epsilon) = (1 - n\epsilon)\lambda_i(F_0) \tag{4.17}$$

The result follows.

□

Therefore we can prove Theorem 4.4.1

Proof. (of Theorem 4.4.1) The Spectral gap gain of F_ϵ with respect to F_0 is

$$\frac{\Delta(F_\epsilon)}{\Delta(F_0)} = \frac{1 - (1 - n\epsilon)\mu(F_0)}{1 - \mu(F_0)} = 1 + \frac{n\epsilon\mu}{1 - \mu}.$$

Therefore in order to get $\frac{\Delta(F_\epsilon)}{\Delta(F_0)} \gg 1$, we should have $\frac{\mu}{1-\mu} \gg \frac{1}{n\epsilon}$. □

Lemma 4.4.2 and Theorem 4.4.1 are important because by utilizing them we can decide whether a given graph or a class of graphs is small-worldizable, what is the spectral gap gain we can get by perturbation, and when (i.e for what values of parameters) the onset of the small world phenomenon occurs. We now consider four different classes of graphs and study whether they are small-worldizable.

- *Disconnected graphs*

  Suppose $\mathcal{G}$ is a graph with p disconnected components $G_0, ..., G_p$. From Perron-Frobenius theory we know that in this case the matrix F_0 has eigenvalue 1 with

multiplicity equal to p. Furthermore, the corresponding Markov chain has p ergodic classes and the perturbation which results in F_ϵ is a singular perturbation. The spectral gap in this case is increased from a zero value to a nonzero value and therefore the spectral gap gain is equal to infinity for any nonzero amount of ϵ. Although trivial, this is an important extreme case, since it distinguishes convergence and non-convergence of consensus algorithms on a graph. The range of small-worldizability for disconnected graph is large ($0 < \epsilon < \frac{1}{n}$) and there is no guaranty that the resulted graph has good convergence properties.

- *Star graphs* The star graph is a centralized topology. There is a central node which is connected to all of the other nodes. On the other hand, all of the other nodes are poorly connected in the sense that they are only connected to central node. Since regardless of the number of nodes, the diameter of any star is only 3 we anticipate that it has high spectral gap. If we denote the central node as node 1, the F_0 matrix can be expressed as:

$$F_0 = \begin{pmatrix} \frac{1}{n} & \frac{1}{n} & \frac{1}{n} & . & . & . & \frac{1}{n} \\ \frac{1}{2} & \frac{1}{2} & 0 & . & . & . & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 & . & . & 0 \\ . & . & . & . & . & . & . \\ \frac{1}{2} & 0 & . & . & . & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & . & . & . & . & \frac{1}{2} \end{pmatrix} \tag{4.18}$$

The eigenvalues of this matrix are

$$\lambda_1 = 1, \qquad \lambda_2 = \lambda_3 = ... = \lambda_{n-1} = \frac{1}{2}, \qquad \lambda_n = -\frac{n-2}{2n}.$$

Therefore $\mu = \frac{1}{2}$ and the spectral gap is also $\frac{1}{2}$. Since the spectral gap of the star topology is non vanishing, it is meaningless to address its small-worldizability; the star is already a 'small world' in the diameter sense. However, it is interesting to study it as an extreme case. This can be also observed by examining the condition of Theorem 1, which needs $\epsilon \gg \frac{1}{n}$ for small-worldizability of a star topology. The highest possible value for spectral gap is 1 for the case of complete graph and taking $\epsilon = \frac{1}{n}$ yields this value. Therefore, we can see that once the initial spectral gap is large enough, there is lesser value in terms of achieving faster convergence for adding extra edges.

- *Hypercube graphs* The hypercube graph Q_m is an $m-$regular graph with $n = 2^m$ vertices, where $\mathcal{V}$ is the set of all n binary $m-$tuples. Two nodes are considered neighbor if the hamming distance is 1. The $n-$dimensional hypercube can be constructed inductively as the following cartesian product; i.e.

$$Q_2 = K_2 \tag{4.19}$$

$$Q_m = K_2 \times Q_m, \tag{4.20}$$

where K_2 is the complete graph on 2 nodes. Figure 4.5 is a hypercube for $n = 16, m = 4$. The spectrum of the adjacency matrix of the Cartesian product of two graphs is known to be the set sum of their spectra [52], i.e. if

$$Spec(G_1) = \{\lambda_i\}_{i=1}^{p}$$

and

$$Spec(G_2) = \{\xi_i\}_{j=1}^{r},$$

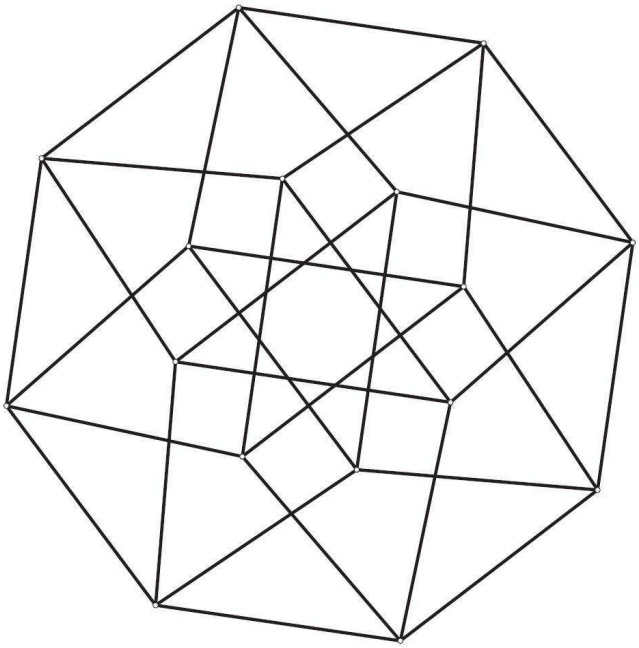


Figure 4.5: Hyper cube (m=4)

then:

$$Spec(G_1 \times G_2) = \{\lambda_i + \xi_j\}_{i=1,j=1}^{p,r}.$$

The eigenvalues of the adjacency matrix of Q_m can be found recursively, and are equal to $m - 2k$ with multiplicity $\binom{m}{k}$, for $k = 0, 1, 2, ..., m$. Since Q_m is $m-$regular, $F_0 = \frac{1}{m+1}(A + I)$, and its eigenvalues are $1 - \frac{2k}{m+1}$ with multiplicity $\binom{m}{k}$, for $k = 0, 1, 2, ..., m$. Therefore, the SLEM of F_0 for a 2^n-hypercube is equal to $\mu = 1 - \frac{2}{m+1} = \frac{\log n - 1}{\log n + 1}$. It follows that $\epsilon = O(\frac{1}{n \log n})$ is the onset of small world effect. The small world effect is dominant for values of ϵ which satisfy $\frac{1}{n \log n} \ll \epsilon \ll \frac{1}{n}$, for example $\epsilon = O(\frac{1}{n \log \log n})$ yields small world effect for a 2^n-hypercube.

- *Higher order lattices* We studied the case of rings in our discussion of Watts-

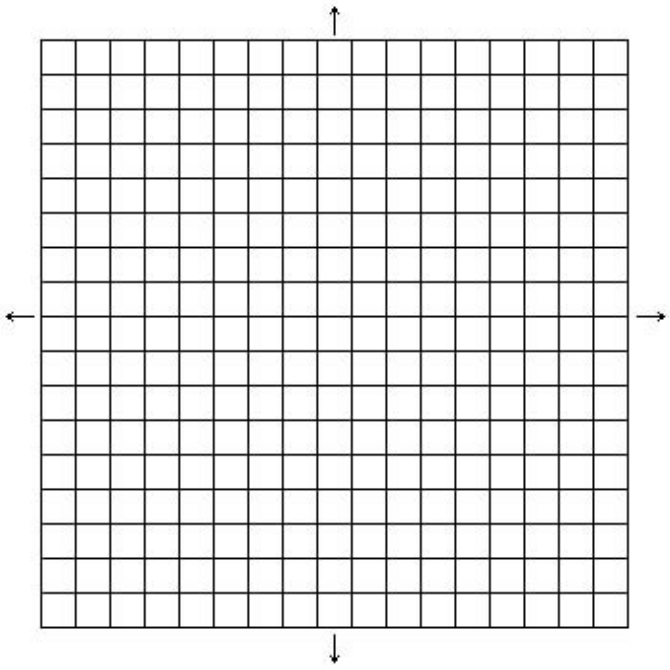


Figure 4.6: Two dimensional grid

Strogatz small world model. The case of higher order lattices yield similar results. Here we present the case of two dimensional square lattices. We consider a two dimensional $m \times m$ grid with periodic boundary (Figure 4.6).

Therefore, we have a graph on $n = m^2$ nodes, in which each node is connected nodes. Using Vicsek's weights, the corresponding matrix is block circulant with each of its blocks being circulant: the matrix F_0 can be written as:

$$F_0 = BlockCirc[C_1, C_2, ..., C_m],$$

Where

$$C_1 = Circ[\frac{1}{5}, \frac{1}{5}, 0, ..., 0, \frac{1}{5}],$$

$$C_2 = C_m = \frac{1}{5} I_m,$$

and

$$C_3 = C_4 ... = C_{m-1} = \mathbf{0}.$$

The matrices C_i are either in diagonal form or can be diagonalized. Let Λ_1 denote the diagonalization of C_1, and $\Lambda_2 = \Lambda_n = \frac{1}{5} I_m$.

The SLEM of F_0 is equal to:

$$\mu(F_0) = \frac{1}{5}[3 + 2cos(\frac{2\pi}{\sqrt{n}})]$$

.

This follows following the fact that as an $m^2 \times m^2$ block circulant matrix whose blocks are m dimensional circulants, the matrix F_0 can be diagonalized using the Kronecker product of m dimensional Fourier as indicated in theorem 5.8 of [30]. Therefore, F_0 can be diagonalized, in which the diagonal matrix Λ is given by:

$$\Lambda = I_m \otimes \Lambda_1 + \Omega_m \otimes \Lambda_2 + \Omega_m^{m-1} \otimes \Lambda_m,$$

where $\Omega_m = diag(1, \omega, ..., \omega^{m-1})$ and $\omega = exp(2\pi i/m)$. Therefore the eigenvalues of m diagonal blocks of λ should be compared to find the SLEM. The result follows by this comparison and the fact that $SLEM(\Lambda_1) = \lambda_2(\Lambda_1)$. Now, consider the perturbed matrix F_ϵ. It follows that $\epsilon = O(\frac{1}{n^2})$ is the onset of small world effect. The small world effect is dominant for values of ϵ which satisfy $\frac{1}{n^2} \ll \epsilon \ll \frac{1}{n}$, for example $\epsilon = O(\frac{1}{n \log n})$ yields small world effect for a two dimensional square lattice with periodic boundary.

4.5 Probabilistic framework

The main focus of the analysis in this chapter has been directed to answer the question of how to characterize small world effect as a perturbation of a base graph. Having answered the the question $[1']$ of section 4.4, we now move on to the other important question of that section:

> Is it possible to increase the convergence rate of the consensus algorithm on a given [sparse] graph substantially by adding a set of few random shortcuts?

To address the above question, we first provide a probabilistic framework for the convergence of consensus problems. This framework enables us to consider the more general problem of determining the convergence speed of consensus problems, in cases where the underlying topology changes probabilistically. This is important on its own right since for example, we find bounds on the convergence speed of probabilistic consensus in the means square sense, which can be used for convergence analysis of consensus type algorithms in uncertain environments. Such applications include consensus type algorithms in wireless setups [116] and local auction based resource allocation on large probabilistic networks [59]. As a byproduct, the model can be considered as switching between multiple topologies to increase the convergence rate. In such an interpretation, communication with remote nodes is done with a very small probability to conserve the node power.

Conditions for convergence of consensus schemes under stochastic frameworks has recently gained attention due to applications which consider link losses and packet drops in the underlying communication topology. Such conditions have been derived, e.g., in

[108, 51, 35, 73] for probabilistic or almost-sure convergence and in [17] for second moment convergence. The rate of convergence was also studied in [17]. We consider the framework of second moment convergence and extend the results of [17] to establish fundamental bounds on the convergence of consensus problems in probabilistically time varying graphs and use the results to address the questions posed at the beginning of this section.

We consider that at each iteration time t, the graph $\mathcal{G}(t)$ can be selected from a finite set

$$\mathbb{G} = \{\mathcal{G}_1, \mathcal{G}_2, \cdots, \mathcal{G}_m\}.$$

We address the convergence results for the case when this selection is performed in an independent and identically distributed (i.i.d.) manner, with the graph $\mathcal{G}_i$ being selected at any time step t with a fixed probability p_i. More complicated models, in which this choice can be carried out according to a underlying Markov chain, can also be analyzed. Proposition 4.5.1 is an example of how our results can be extended to such cases.

To make the problem of determining the convergence rates more tractable, we consider the *average consensus* problem. Using doubly stochastic weight matrices results in average consensus i.e, the nodes converge to the average of the initial values $x_i(0)$. The system thus evolves according to the discrete time equation

$$x(t+1) = F(t)x(t), \qquad x(0) = x_0, \tag{4.21}$$

where F is a doubly stochastic matrix. Furthermore, we use the symmetric weights derived from the Laplacian of the graphs, i.e. we use the class of weight matrices

$$F(t) = I - hL(t),$$

where h is a small positive parameter and $L(t)$ corresponds to the Laplacian of the graph $\mathcal{G}(k)$.

This class of weight matrices can be considered as a result of discretizing the continuous time consensus problems of the type considered by Olfati-Saber and Murray in in [86].

We denote the symmetric consensus matrix corresponding to the graph $\mathcal{G}_i$ by $F_i = I - hL_i$ and denote the set of F_is corresponding to $\mathbb{G}$ by $\mathbb{F}$. Since the state $x(t)$ now evolves stochastically, the convergence of the node values to the average consensus value occurs in a probabilistic fashion. We are interested in second moment convergence, i.e., the convergence of the covariance of the state vector $x(t)$ to its final value $\mathbf{1}\mathbf{1}^T\mu^2$, where μ is the average of the initial values $x_i(0)$. This is equivalent to studying the convergence of the vector $x(t)$ to its final value $\mu\mathbf{1}$. To see this, consider the following Lyapunov function candidate:

$$V(x(t)) = \frac{1}{n}[\sum_{i\neq j} E(||x_i(t) - x_j(t)||^2)] = E[x^T(t)(I - \frac{\mathbf{1}\mathbf{1}^T}{n})x(t)] = E[x^T(t)\hat{L}x(t)], \tag{4.22}$$

where $\hat{L} = (I - \frac{\mathbf{1}\mathbf{1}^T}{n})$ can be viewed as the Laplacian of the complete graph. $\hat{L}$ is a projection, i.e., $\hat{L}^2 = \hat{L}$. Let

$$P(x(t)) = E[\hat{L}x(t)x(t)^T\hat{L}],$$

so that $V(x(t)) = Tr(P(x(t)))$. Convergence of the vector $x(t)$ to $\mu\mathbf{1}$ is equivalent to each entry of P converging to zero. Finally, define $G(t)$ to be the covariance of $x(t)$, i.e., $G(t) = E[x(t)x^T(t)]$. $P(x(t))$ and $G(t)$ are related via

$$P(x(t)) = \hat{L}G(t)\hat{L}.$$

Convergence of $P(x(t))$ to zero is equivalent to convergence of $G(t)$ to a constant matrix. The two notions of mean square convergence mentioned above are identical.

The following result determines the conditions for convergence and the rate of convergence. The results about the i.i.d. case have been presented before in [17].

Proposition 4.5.1. *Consider the consensus algorithms of Section 4.5, however, with the consensus matrix being chosen from the set $\mathbb{F}$ either in an i.i.d. or according to a Markov chain with transition probability matrix Q.*

1. *For the i.i.d. case, the system converges in the second moment sense if*

$$\rho\left(E[F \otimes F] - \frac{1}{n^2}\mathbf{1}\mathbf{1}^T\right) < 1,$$

where $\rho(.)$ denotes the spectral radius and $\otimes$ denotes the Kronecker product. Further, the rate of convergence is governed by $\rho\left(E[F \otimes F] - \frac{1}{n^2}\mathbf{1}\mathbf{1}^T\right)$ or the SLEM of the matrix $E[F \otimes F]$.

2. *For the Markovian case, the system converges in the second moment sense if*

$$\rho\left((Q^T \otimes I)\,(diag(E[F_i \otimes F_i])) - \frac{1}{n^2}\mathbf{1}\mathbf{1}^T\right) < 1,$$

where $diag(A_i)$ denotes a block diagonal matrix with blocks A_i. Further, the rate of convergence is governed by $\rho\left((Q^T \otimes I)\,(diag(E[F_i \otimes F_i])) - \frac{1}{n^2}\mathbf{1}\mathbf{1}^T\right)$ or the SLEM of the matrix $(Q^T \otimes I)\,(diag(E[F_i \otimes F_i]))$.

Proof. The proof follows immediately if we consider the evolution of the covariance $G(t)$. We have

$$\begin{aligned} G(t+1) &= E[x(t+1)x^T(t+1)] \\ &= E[F(t)x(t)x^T(t)F(t)]. \end{aligned} \tag{4.23}$$

For the i.i.d. case, vectorize the equation to yield

$$\text{vec}(G(t+1)) = \text{vec}(E[F(t)x(t)x^T(t)F(t)]).$$

Since the matrix choice at t is independent of $F(t)$,

$$\text{vec}(G(t+1)) = E[F(t) \otimes F(t)]\text{vec}(E[x(t)x^T(t)]),$$

where the first expectation is over the choice at t. Thus

$$\text{vec}(G(t+1)) = E[F \otimes F]\text{vec}(G(t)).$$

Since each F_i is stochastic, so is $E[F \otimes F]$ and thus convergence is governed by its second largest eigenvalue modulus (SLEM). This proves the theorem for the i.i.d. case.

The proof for the Markovian case is similar. Denote the state of the Markov chain at time t by $r(t)$. Define the conditional covariance

$$G_i(t) = E[x(t)x^T(t)|r(t) = i]\text{Prob}(r(t) = i).$$

$G(t)$ is obtained by summing over all $G_i(t)$. We have

$$\begin{aligned} G_i(t+1) &= E[F(t)x(t)x^T(t)F(t)|r(t+1) = i].\text{Prob}(r(t+1) = i) \\ &= \sum_{j=1}^{m} E[F(t)x(t)x^T(t)F(t)|r(t+1) = i, r(t) = j]. \\ &\quad \text{Prob}(r(t) = j|r(t+1) = i)\text{Prob}(r(t+1) = i). \end{aligned}$$

Since $x(t)$ and $r(t+1)$ are conditionally independent given $r(k)$, we can vectorize the equation to obtain

$$\text{vec}(G_i(t+1)) = \sum_{j=1}^{m} E[F_j \otimes F_j]\text{vec}(G_j(t))\text{Prob}(r(t+1) = i|r(t) = j),$$

where we have used Bayes rule. On stacking $G_i(t)$ for all i, we obtain that the recursion is governed by the matrix $(Q^T \otimes I)\,(\text{diag}(E[F_i \otimes F_i]))$. Since this matrix is also stochastic, the rate is governed by its SLEM. □

The result given above characterizes the rate of convergence of the second moment in the average consensus protocol. We may note that since the rate of convergence has not yet been calculated in a closed form for arbitrary graphs even for static topologies, we cannot expect to compute the rates while switching over arbitrary graphs. Moreover, the rate of convergence for probabilistic consensus protocols depends on the topology in a complicated way. Thus, for further analytic results and insights, we will concentrate on graphs with certain amount of symmetry.

The calculation of the SLEM is particularly difficult since, in general, calculating the Kronecker product requires $n^2 \times n^2$ matrix operations for n agents. Because of the presence of the expectation operator, even for symmetric graphs, the eigenvalue calculations can quickly become complicated. This complexity has also been recognized, e.g., [17] where instead other metrics are used as a proxy for such eigenvalues. In contrast, we will continue to focus on the SLEM of these matrices by calculating lower and upper bounds on the rate of convergence. We will show in the next two sections, that such bounds can yield interesting insights. To make the problem more tractable, we will impose a condition which results in consensus matrices with nonnegative eigenvalues. Therefore the SLEM of the matrix will be the second largest eigenvalue. The following lemma provides this condition:

Lemma 4.5.2. *If $h < \frac{1}{2d_{max}(\mathcal{G})}$, all the eigenvalues of $I - hL(\mathcal{G})$ are nonnegative.*

Proof. From Gershgorin's circle theorem [106], every eigenvalue of L should lie in one of the circles with L_{ii} as center and radius equal to $\sum_{j\neq i}|L_{ij}|$. However, L_{ii} and $\sum_{j\neq i}|L_{ij}|$ both equal the degree of node i in graph $\mathcal{G}$. Let λ_1 denote the maximum eigenvalue of the Laplacian L. Gershgorin's theorem results that $\lambda_1 \leq 2d_{max}$, where d_{max} denotes the maximum degree of graph $\mathcal{G}$. The smallest eigenvalue of $I - hL$ is $1 - h\lambda_1$. The result follows from the fact that $1 - h\lambda_1 > 1 - 2hd_{max}$, and the assertion that $2hd_{max} < 1$. □

The following proposition provides the upper and lower bounds for the SLEM of the consensus matrices.

Proposition 4.5.3. *Denote $A = E[F \otimes F]$, where $F = I - hL$. Also denote the average value of the Laplacian $E[L]$ by $\bar{L}$. Finally, let λ_i be the i^{th} largest eigenvalue of a Laplacian matrix L,*[1]*. i.e.*

$$\lambda_1(\bar{L}) \geq \lambda_2(\bar{L}) \geq ... \geq \lambda_n(\bar{L}).$$

Then,

$$1 - h\lambda_{n-1}(\bar{L}) \leq \lambda_2(A) \leq 1 - h\lambda_{n-1}(\bar{L}) + h^2\lambda_1(E[L \otimes L]).$$

To prove this result, we use results from matrix perturbation theory, see, e.g., [104], Theorem 4.8 and Corollary 4.9:

Theorem 4.5.4. *Let A_0 be a Hermitian matrix with eigenvalues*

$$\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_n$$

[1]Usually the i^{th} smallest eigenvalue of the Laplacian is denoted by λ_i in the literature. We have not followed the convention, to be consistent with our choice of ordering of eigenvalues of the weight matrices F.

and $A = A_0 + E$ be a Hermitian perturbation of A with eigenvalues

$$\tilde{\lambda}_1 \geq \tilde{\lambda}_2 \geq ... \geq \tilde{\lambda}_n$$

Furthermore, let the eigenvalues of E be

$$\epsilon_1 \geq \epsilon_2 \geq ... \geq \epsilon_n.$$

Then for $i = 1, 2, ..., n$

$$\tilde{\lambda_i} \in [\lambda_i + \epsilon_n, \lambda_i + \epsilon_1]. \tag{4.24}$$

Using this result, we prove Proposition (4.5.3).

Proof. Let

$$\begin{aligned} A &= E[(I - hL) \otimes (I - hL)] \\ &= I - hI \otimes \bar{L} - h\bar{L} \otimes I + h^2 E[L \otimes L] \\ A_0 &= I - hI \otimes \bar{L} - h\bar{L} \otimes I \\ E &= h^2 E[L \otimes L]. \end{aligned}$$

The Kronecker product of the Laplacian has the smallest eigenvalue equal to zero. $E[L \otimes L]$ preserves the property that its row sums all equal zero. Therefore, $\epsilon_n = 0$. Furthermore,

$$\lambda_2(A_0) = \lambda_2(I - hI \otimes \bar{L} - h\bar{L} \otimes I) = 1 - h\lambda_{n-1}(\bar{L}).$$

Therefore

$$\lambda_2(A) \in [1 - h\lambda_{n-1}(\bar{L}), 1 - h\lambda_{n-1}(\bar{L}) + h^2\lambda_1(E[L \otimes L])].$$

□

The result given by proposition (4.5.3) indicates that for finding bounds on the convergence rate of probabilistic consensus algorithms on a set of matrices $\mathbb{F}$, we should

1. Find the exact value or bounds for $\lambda_{n-1}(E[L])$,

2. Find the exact value or bounds for $\lambda_1(E[L \otimes L])$.

As the examples in the next section will show $\lambda_{n-1}(E[L])$ can be computed for many different classes of graphs. To find bounds on $\lambda_1(E[L \otimes L])$, we use the fact that all the matrices L that we consider are symmetric and positive semi-definite. For such matrices, the spectral radius $\lambda_1(.)$ is a convex function. Thus, Jensen's inequality can be applied to obtain

$$\lambda_1(E[L \otimes L]) \leq E(\lambda_1([L \otimes L])) = E[(\lambda_1(L))^2].$$

Since all the eigenvalues of a graph Laplacian are bounded by twice the maximum degree of the graph, we obtain

$$\lambda_1(E[L \otimes L]) \leq 4E[d_{max}^2].$$

The upper bound on the spectral gap is $h\lambda_{n-1}(\bar{L})$. We may need to change the amount of h when we use the switching scheme to remain consistent with the change in the degree of graph nodes due to switching. This change is however not very significant, when the switching probability is low enough, as seen in the examples of the next sections. We develop a necessary condition for a graph to improve convergence rate as a result of uniform probabilistic switching.

Consider a given graph $G(V, E)$, with Laplacian L_0. Let its complement graph be denoted by $G^c = (V, E^c)$, where $E^c = \{e | e \notin E\}$. Consider a uniform switching in

which all the edges of E^c can be used with a small probability $0 < \epsilon < 1$. Then the expected Laplacian is:

$$\bar{L} = L_0 + \epsilon L_0^c$$

Notice that $L_0 + L_0^c = nI - \mathbf{1}\mathbf{1}^T$, and that the vector $\lambda_n(\bar{L}) = \mathbf{1}$ is an eigenvector of L_0^c with the corresponding eigenvalue 0. Taking a set of orthogonal eigenvectors, it can be easily verified that for $1 \leq i \leq n-1$,

$$\lambda_i(L_0^c) = n - \lambda_{n-i}(L_0).$$

Therefore, we have the following result:

Corollary 4.5.5. *A necessary condition for getting significant rate improvement by uniform switching is that $\lambda_1(L_0) \ll n$, where $\lambda_1(L_0)$ is the spectral radius of L_0.*

Proof. We know that $\bar{L}$ is a positive semi-definite matrix and since $\bar{L}\mathbf{1} = \mathbf{0}$, we get $\lambda_n(\bar{L}) = 0$. Furthermore if for $1 \leq i \leq n-1$, the vector x_i is the eigenvector corresponding to $\lambda_i(L_0)$, then:

$$\bar{L}x_i = (L_0 + \epsilon L_0^c)x_i = [\lambda_i(L_0) + \epsilon(n - \lambda_{n-i}(L_0))]x_i,$$

Therefore $\lambda_{n-1}(\bar{L}) = \lambda_{n-1}(L_0) + \epsilon(n - \lambda_1(L_0))$, and this proves the statement of the corollary. □

4.6 Watts-Strogatz small world graphs

We now return to our analysis of small world graphs. We consider the nominal base graph to be a ring and the phi-model as in Section 4.2. To model the existence

of a few long range links, we assume that, at each time, each agent can establish a link with non-adjacent nodes with a small probability ϵ. By using the probabilistic framework developed in Section 4.5, we now analyze the effect of such long range links, established with a small probability, on the convergence rate. In particular, we assume that $\epsilon \propto n^{-\alpha}$ where as before n is the number of nodes and α is a natural number.

The consensus matrix of the nominal ring graph is:

$$I - hL_{fix} = \begin{bmatrix} 1-2h & h & 0 & ... & h \\ h & 1-2h & h & ... & 0 \\ . & . & . & . & . \\ h & 0 & ... & h & 1-2h \end{bmatrix}.$$

The SLEM for this graph can be easily calculated using the circulant structure of the matrix as in Section 4.3

$$\lambda_2(F_0) = 1 - 2h + 2h\cos(\frac{2\pi}{n}).$$

Thus, the spectral gap of the nominal graph is given by

$$S.G.(fixed) = 2h(1 - \cos(\frac{2\pi}{n})).$$

Now we analyze the effect of the additional links. To calculate the expected Laplacian matrix, we realize that

$$\bar{L}_{ij} = \begin{cases} 2 + (n-3)\epsilon & i = j \\ -1 & |i-j| = 1 \\ -1 & (i,j) = (1,n) \text{ or } (i,j) = (n,1) \\ -\epsilon & \text{otherwise.} \end{cases}$$

This graph has a circulant structure. Thus,

$$\lambda_{n-1}(\bar{L}) = 2 + (n-3)\epsilon - 2(1-\epsilon)\cos(\frac{2\pi}{n})$$

Using Proposition 4.5.3 we thus obtain the following bounds when long range links are added with a small probability ϵ.

$$\begin{aligned}
&\lambda_2(E[(I - hL) \otimes (I - hL)]) \in \\
&[1 - h(2 + (n-3)\epsilon - 2(1-\epsilon)\cos(\frac{2\pi}{n})), \\
&1 - h(2 + (n-3)\epsilon - 2(1-\epsilon)\cos(\frac{2\pi}{n})) + 4h^2 E[d_{max}^2]].
\end{aligned}$$

The spectral graph for this case evaluates to

$$\begin{aligned}
&S.G.(Switching) \in \\
&[h(2 + (n-3)\epsilon - 2(1-\epsilon)\cos(\frac{2\pi}{n})) - 4h^2 E[d_{max}^2], \\
&h(2 + (n-3)\epsilon - 2(1-\epsilon)\cos(\frac{2\pi}{n}))]
\end{aligned}$$

For $\epsilon = n^{-\alpha}$ and $\alpha = 1, 2, 3, ...$ we can use a Chernoff bound argument similar to that outlined in Lemma 3 of [107] to show that in the limit of large n, $d_{max} < \log n$ almost surely. Note that d_{max} can never exceed n.

We are now ready to compare the spectral gaps of the nominal topology and of the one with long range links. We assume $h \propto n^{-1}$ in keeping with our assumption relating h to the maximum degree. We note the following observations, which are similar in form to the results of section 4.3.

1. As the number of nodes n increases, the spectral gap for the fixed graph varies as n^{-3}.

2. The upper bound for the spectral gap for the graph with large range links evaluates to

$$h[n^{1-\alpha} - n^{-\alpha} + 2\pi^2 n^{-2} - 2n^{-\alpha-2}] = o(n^{-\alpha}).$$

Thus, for $\alpha \geq 3$ the dominant term is also the n^{-3} term. Thus even if we consider the upper bound, the presence of long range links cannot increase the spectral gap if the links are added with too small a probability.

3. The lower bound on the spectral gap of the case with long range links, for large n is approximately

$$h[n^{1-\alpha} - n^{-\alpha} + 2\pi^2 n^{-2} - 2n^{-\alpha-2}] - \frac{(\log n)^2}{n^2}.$$

For $\alpha = 1$, this bound evaluates to $o(n^{-1})$, which is an order of magnitude better than the nominal ring case. This shows the huge impact of long range links, even if they are added with a vanishingly small probability.

4. If we take $\alpha = 2$ then the lower bound is not tight enough to make any statement about the comparison of the two regimes.

From the above observations, we can conclude that long range links can improve the convergence rate of the consensus protocols enormously. The probability of adding long range links is an indicator of the number of long range links in the deterministic model. As we increase the probability, we get a sharp increase in the convergence rate at $\alpha = 2$. This improvement in performance can be viewed as the consequence of the onset of small world phenomena.

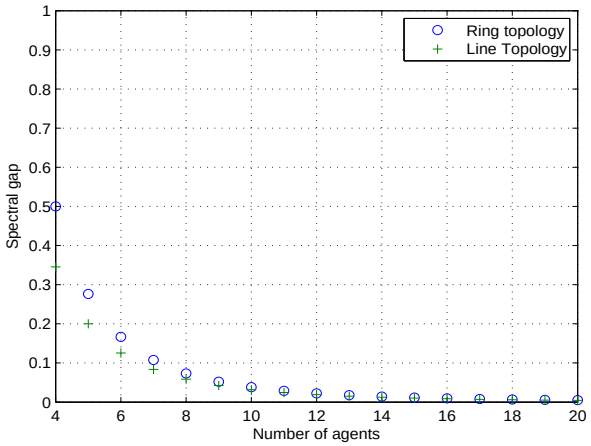


Figure 4.7: Spectral gap for a ring and a line topology.

Similar expressions for rings in higher dimensions can be obtained. As stated above, for general graphs, it may not be possible to prove the increase in rate due to extra edges analytically. However, some graphs of practical concern, such as regular lattices, can be well-approximated by rings, as the number of agents increases. As an example, Figure 4.7 shows the spectral gap for a ring and a line topology. Spectral gaps are quite similar for a fairly small number of agents. Similarly, Figure 4.8 shows the lower and upper bounds for the spectral gap when edges are added with $\alpha = 1$. It can be seen that the bounds for the ring and the line topologies match for a fairly small number of agents. Also the upper and lower bounds are quite close to each other for a moderate number of agents. Thus, the bounds seem quite tight, at least for this example.

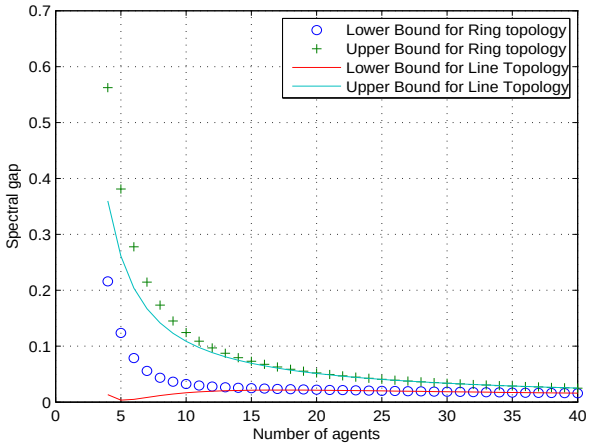


Figure 4.8: Upper and lower bounds for spectral gap for a ring and a line topology.

4.7 Some Other Scenarios

The framework, methods and tools developed and extended in Section 4.5 and Section 4.6 here, can be used to analyze performance of the average consensus protocol when probabilistic switching occurs due to any reason. In this section, we demonstrate this using various simple scenarios.

4.7.1 Topology Switch due to Changing Neighbors

Consider n agents placed on a ring with n empty slots. Being on a ring constrains each node's neighbors to the agent to its left and the agent to its right. We consider a protocol in which at every time step an agent chooses two neighbors randomly. This can be viewed as a variation of the Gossip Algorithm proposed in [17]. We model the selection of neighbors by assuming that at every time step, each agent chooses a slot at random and with equal probability among all the possibilities. Moreover, every slot contains only one

node at each time. This is equivalent to the assumption that at each time agents randomly choose their neighbors bi-directionally, while constraining the total number of neighbors to two.

We wish to compare the rate of convergence of this scheme with the nominal case in which there is a fixed ring topology. For the fixed ring topology G_0, the consensus matrix is again circulant. Thus, the SLEM is equal to $\lambda_2 = 1 - 2h\cos(\frac{2\pi}{n})$. The spectral gap, thus, is given by

$$S.G.(fixed) = 2h[1 - \cos(\frac{2\pi}{n})] = 4h\sin^2(\frac{\pi}{n}).$$

For large n this is approximately $\frac{4\pi^2 h}{n^2}$ and varies as n^{-2}.

For the case when topology switch occurs, we need to compute the expected value of the Laplacian matrix. We use the fact that $\bar{L}_{ii}$ equals the expected number of agent i's neighbors, which is 2 in this case. Moreover, for the non-diagonal terms, $-\bar{L}_{ij}$ is equal to the probability that the agents i and j are neighbors. Therefore, $\bar{L}$ has the structure

$$\bar{L} = \begin{bmatrix} 2 & -\frac{2}{n-1} & \cdots & -\frac{2}{n-1} \\ -\frac{2}{n-1} & 2 & \cdots & -\frac{2}{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ -\frac{2}{n-1} & -\frac{2}{n-1} & \cdots & 2 \end{bmatrix}. \tag{4.25}$$

By exploiting the circulant structure, it is seen that

$$\lambda_{n-1}(\bar{L}) = 2 + \frac{2}{n-1}.$$

Therefore,

$$\lambda_2(I - h\bar{L}) = 1 - 2h - \frac{2h}{n-1}.$$

Finally, to calculate the upper bound, notice that the degree of each node is 2 with probability 1. Thus, using Proposition 4.5.3, the bounds on the spectral gap are

$$h(2+\frac{2}{n-1})-16h^2 \leq S.G.(switching) \leq h(2+\frac{2}{n-1})$$

Thus, it can be seen that

$$\frac{S.G.(fixed)}{S.G.(switching)} \leq \frac{4h\sin^2(\frac{\pi}{n})}{2h+\frac{2h}{n-1}-16h^2}$$

For the limit of large n, assuming a constant $h < \frac{1}{8}$ (as dictated by the maximum degree of any node), the numerator varies as n^{-2} whereas the denominator varies as n^{-1}. Thus, the ratio approaches zero with increasing n. This shows that even the lower bound of the spectral gap of the switching case shows order of magnitude improvement compared to the spectral gap of the fixed topology.

This remarkable increase in rate of convergence by switching may yield the conjecture that switching to far away neighbors always increases the rate, provided that the switching is done with a high enough probability. This conjecture, as stated above, is, however, false, as shown in the next subsection.

4.7.2 Erdos-Renyi Random graphs

In this sub-section we consider a case where switching to far away neighbors does not increase the rate of convergence of the consensus protocol. This case also yields the rate of convergence for the class of random graphs known as Erdos-Renyi random graphs. Once again, consider n nodes to be present. Suppose that there exists a link between any two nodes with probability $q \in (0,1]$. The existence of a link between any two nodes is

therefore random and independent of the other connections. We compare the convergence rates between two cases.

1. Fixed random graph: The choice of the links is done (randomly) at time 0 and the graph stays constant after that.

2. Switched random graph: The choice of the links is done at every time step. We assume that the choice of the links is done independently with respect to time. Thus, the random graph at each time is independent of the choice of the random graph at other time instants.

For the fixed random graph case, we use a high probability bound due to Fiedler, reported in [60, 51] on the second smallest eigenvalue of the Laplacian of a random graph. For the limit of large n and for $\epsilon \in (0, 2)$, we obtain

$$\lim_{n\to\infty} Pr\{qn - \sqrt{(2+\epsilon)q(1-q)n\log n} <$$

$$\lambda_{n-1}(L(G(n,q))) < qn - \sqrt{(2-\epsilon)q(1-q)n\log n}\} = 1.$$

Therefore if a fixed random topology is used at all times, then, with high probability

$$1 - hqn + h\sqrt{(2-\epsilon)q(1-q)n\log n} < \lambda_2[I - hL_{fixed}] <$$

$$1 - hqn + h\sqrt{(2+\epsilon)q(1-q)n\log n}.$$

For the switched random graph case, we calculate the bounds given by Proposition 4.5.3. If at different times we switch between different random graphs, then $\bar{L}_{ii} = E[deg(i)]$, where $deg(i)$ denotes the degree of the node i. To calculate this, denote by X_{ij} the indicator of existence of a link between nodes i and j. Owing to the independence of the links between various nodes, X_{i1}, X_{12}, $\cdots$ are independent Bernoulli random variables. Therefore $\bar{L}_{ii} = q(n-1)$. We can also calculate $\bar{L}_{ij} = -q$. Hence, the expected

value of the Laplacian matrix is given by

$$\bar{L} = q \begin{bmatrix} n-1 & -1 & ... & -1 \\ -1 & n-1 & ... & -1 \\ . & . & ... & . \\ -1 & -1 & ... & n-1 \end{bmatrix}$$

This is again circulant and thus $\lambda_{n-1}(\bar{L}) = qn$. Finally, using Proposition 4.5.3, we obtain the lower bound as

$$\lambda_2(E[(I - hL) \otimes (I - hL)]) > 1 - hnq.$$

One interesting regime to consider is when $q = \Theta(\frac{\log n}{n})$. In this regime, with high probability, a random graph is connected [15]. For a given large n, taking $q = k\frac{\log n}{n}$ with $k \geq 2$, yields

$$S.G.(fixed) < hqn - h\sqrt{(2+\epsilon)q(1-q)n\log}$$

$$\sim (k - \sqrt{2k})h\log n$$

On the other hand for the switching case we have:

$$S.G.(Switching) > hnq = kh\log(n)$$

Both bounds are of the same order. This indicates that there is no large improvement in the rate of convergence in this regime through switching. We conjecture that for switching to help increase the convergence rate, it is not enough to connect long range neighbors. Instead, it seems that the expected diameter of the graph through switching should be much smaller than that of the fixed graph, for switching to be useful.

4.7.3 IID Link Losses due to Communication Failures

Another important case that can lead to unintended switching between topologies is when the communication links can be modelled according to an analog erasure model. In this model, at each time step, a link is functional with a certain probability. We assume that the failures for any particular link occur independently across time and with respect to other link failures. Moreover, we assume that the link failures are bi-directional. Finally, for ease of presentation, we also assume here that each link fails with the same probability p at any time step.

We illustrate the use of our results on a 1-D lattice with a periodic boundary condition, i.e., a ring. The spectral gap of a ring when no link losses occur is given by

$$S.G.(fixed) = 2h(1 - \cos\frac{2\pi}{n}).$$

If link losses occur, the average Laplacian $\bar{L}$ is given by

$$L_{ij} = \begin{cases} 2(1-p) & i = j \\ -(1-p) & |i-j| = 1 \\ -(1-p) & (i,j) = (1,n) \text{ or } (i,j) = (n,1) \\ 0 & \text{otherwise.} \end{cases}$$

Thus, the second smallest eigenvalue is given by

$$\lambda_{n-1}(\bar{L}) = 2(1-p)(1 - \cos\frac{2\pi}{n}).$$

Finally, the spectral gap due to link losses decreases at least by

$$\frac{S.G.(fixed)}{S.G.(losses)} \geq \frac{1}{1-p}.$$

4.8 Design of robust Communication topologies

We now consider the second basic problem which was considered in section 4.1: designing graph topologies which are robust to link losses. As indicated in section 3.5.2 we consider $\tau(\mathcal{G})$ as a measure of system robustness to changes in communication topology. We first address the design problem of section 4.1, where the performance measure to be optimized is the number of spanning trees in a graph $\tau(\mathcal{G})$. Then we show that small world networks are efficient networks in the sense that they are resilient to link losses and provide heuristics for design of optimal topologies.

4.8.1 Problem statement

Given an initial graph topology, we want to add k edges, so that the resulting graph topology has the maximal number of spanning trees among all possible topologies. Consider the design problem of section 4.1 with $\tau(\mathcal{G})$ as the performance measure.

The dynamic graph evolution is given by:

$$\begin{aligned} &G(t+1) = Add(G(t), u(t)), \quad t = 0, 1, ...k-1 \\ &u(t) = e(t+1), \qquad\qquad\qquad e(t+1) \in S \subseteq E(\bar{G}(t)) \\ &G(t) = \mathcal{G}_0 \end{aligned} \tag{4.26}$$

The problem is to:

$$\begin{aligned} \text{maximize} \quad & \tau(G(t+k)) \\ \text{subject to:} \quad & (4.26) \end{aligned} \tag{4.27}$$

where $\tau(G(t) = \prod_{i=1}^{n-1} \lambda_i(L)$.

We relax the problem using a framework similar to [38] Consider $G(1) = Add(\mathcal{G}_0, (l, p))$, then we can write:

$$L(1) = L(0) + (e_l - e_p)(e_l - e_p)^T,$$

where $L(i) \triangleq L(G(i))$. By indexing all candidate edges from 1 to m, denoting the corresponding incident vectors by $f_i = (e_{i_1} - e_{i_2})$ where i_1 and i_2 are the two ends of a candidate edge i, and introducing binary valued variables

$$x_i = \begin{cases} 1, & \text{if edge i is chosen,} \\ 0, & \text{otherwise,} \end{cases}$$

we can write equation (4.27) as:

$$\begin{aligned} &\text{maximize} \quad \tau(L(0) + \textstyle\sum_{i=1}^{l} x_i f_i f_i^T) \\ &\text{subject to:} \quad 1^T x = k \\ &x \in \{0,1\}^m. \end{aligned} \tag{4.28}$$

We now relax the above problem. Let $x > 0$. Consider $F_1(x) = [n\tau\left(L(x)\right)]^{\frac{1}{n-1}} = \left(\prod_{i=1}^{n-1} \lambda_i\right)^{\frac{1}{n-1}}$ and $F_2(x) = \log\det(L(x)) = \log\prod_{i=1}^{n-1} \lambda_i$, which have the same maximizers.

Both of the above functions are concave functions of x for $x > 0$. This is because, for example $g(L) = \det\log(L + \frac{J}{n})$ is a concave function for its positive definite argument L, and $L(x) = L(0) + \frac{1}{n}J + \sum_{i=1}^{l} x_i f_i f_i^T$ is an affine function of x. Therefore the composition $g \circ L$ is a concave function. We have the relaxed convex problem:

$$\begin{aligned} &\text{maximize} \quad \left[F_1(x) = \prod_{i=1}^{n-1} \lambda_i(L(x))\right]^{\frac{1}{n-1}} \\ &\text{subject to:} \quad 1^T x = k \\ &x > 0, \end{aligned} \tag{4.29}$$

or equivalently,

$$\begin{aligned} &\text{maximize} \quad F_2(x) = \log\det(L(x)) \\ &\text{subject to:} \quad 1^T x = k \\ &x > 0. \end{aligned} \tag{4.30}$$

We consider maximizing $F_2(x)$. The first order optimality condition requires that for maximum point $x^\star$,

$$\nabla F_2(x^\star)^T(x - x^\star) \leq 0$$

should hold for all $x > 0$ for which $1^T x = k$. Following [10], if $x_i^\star$ is a nonzero entry and j is an arbitrary index with $j \neq i$, selection of x such that

$$x_k = \begin{cases} x_k^\star, & \text{if } k \neq i, j \\ 0, & \text{if } k = i \\ x_i^\star + x_j^\star & \text{if } k = j, \end{cases} \tag{4.31}$$

yields that at the maximum point $x^\star$ for all $j = 1, ..., m$,

$$\frac{\partial F_2(x^\star)}{\partial x_i} \geq \frac{\partial F_2(x^\star)}{\partial x_j},$$

Therefore at $x^\star$, $F_2(x)$ has equal derivative with respect to all positive x_i.

Taking the derivative yields that at the maximum, for all $x_i > 0$,

$$\begin{aligned} \text{Trace}\left((L_0 + \frac{1}{n}J + \sum_{i=1}^{m} x_i f_i f_i^T)^{-1} f_i f_i^T\right) &= f_i^T (L(x) + \frac{1}{n}J)^{-1} f_i \\ &= \lambda > 0. \end{aligned} \tag{4.32}$$

The term $f_i^T(L(x)+\frac{1}{n}J)^{-1}f_i$ is equal to the effective resistance (distance) between the two ends of the potential edge f_i. Since $F_2(x)$ is a concave function on a convex domain, the optimality conditions are also sufficient. Therefore, If feasible, one should add edges in a way that the effective resistance distance of all selected edges become equal. Also, the selected edges should be between the nodes with the highest resistance difference. Since it is not always possible to add the edges in this way, a good heuristic should make the difference between the effective resistance of the candidate edges as small as possible. We now address special cases of adding one or two edges, which provide more insight on how adding edges increases the number of spanning trees.

4.8.2 Adding one or two edges to a general graph

Consider adding an edge to a general initial graph, $G(0) = \mathcal{G}_0$, which results in a new graph, $G(1)$. As before enumerate the nodes of the graph from 1 to n. The following result holds.

Theorem 4.8.1. *The optimal edge is between two nodes with maximal effective resistance distance.*

Proof. Take two previously disconnected nodes $\alpha, \beta \in \{0, 1, ..., n\}$, and connect them by an edge. The incidence vector for this edge is $f = e_\alpha - e_\beta$. The number of spanning trees in $G(1)$ is:

$$\tau(G(1)) = \frac{1}{n}\det\left(L + \frac{1}{n}J + (e_\alpha - e_\beta)(e_\alpha - e_\beta)^T\right)$$
$$= \frac{1}{n}\left(1 + (e_\alpha - e_\beta)^T(L + \frac{1}{n}J)^{-1}(e_\alpha - e_\beta)\right) det(L + \frac{1}{n}J)$$
$$= \left(1 + (e_\alpha - e_\beta)^T(L + \frac{1}{n}J)^{-1}(e_\alpha - e_\beta)\right)\tau(\mathcal{G}_0) \tag{4.33}$$

If we denote $Z = (L + \frac{1}{n})^{-1}$, then

$$\tau(G(1)) = (1 + Z_{\alpha\alpha} - 2Z_{\alpha\beta} + Z_{\beta\beta})\tau(\mathcal{G}_{(0)})$$
$$= (1 + R_{eff}(\alpha, \beta))\,\tau(\mathcal{G}_{(0)}) \tag{4.34}$$

Therefore, adding an edge between two nodes with the highest effective resistance distance results in the highest increase in the number of spanning trees of any general graph. □

We now consider addition of two edges (α, β) and (γ, δ) to the initial graph $G^{(0)}$.

$$G(2) = Add\left(Add(\mathcal{G}_0, (\alpha, \beta)), (\gamma, \delta)\right).$$

The corresponding incidence vectors for the edges are, $f_{\alpha\beta} = e_\alpha - e_\beta$ and $f_{\gamma\delta} = e_\gamma - e_\delta$. Also, as before we let $Z = (L + \frac{1}{n}J)^{-1}$. Then,

$$\tau(G(2)) = \frac{1}{n}\det\left(L + \frac{1}{n}J + f_{\alpha\beta}f_{\alpha\beta}^T + f_{\gamma\delta}f_{\gamma\delta}^T\right)$$
$$= \frac{1}{n}\left(1 + f_{\gamma\delta}^T(L + \frac{1}{n}J + f_{\alpha\beta}f_{\alpha\beta})^{-1}f_{\gamma\delta}\right)$$
$$.\det(L + \frac{1}{n}J + f_{\alpha\beta}f_{\alpha\beta}^T)$$
$$= \frac{1}{n}\left(1 + f_{\gamma\delta}^T(Z - (1 + f_{\alpha\beta}^T Z f_{\alpha\beta})^{-1}Z f_{\alpha\beta}f_{\alpha\beta}^T Z)f_{\gamma\delta}\right)$$
$$.\det(L + \frac{1}{n}J + f_{\alpha\beta}f_{\alpha\beta}^T), \tag{4.35}$$

where the third equality follows from the Sherman-Morrison-Woodbury formula for the inverse of a rank one modification of a matrix. In equation (4.35), $\det(L+\frac{1}{n}J+f_{\alpha\beta}f_{\alpha\beta}^T)$, $f_{\alpha\beta}^T Z f_{\alpha\beta}^T$, and $f_{\gamma\delta}^T Z f_{\gamma\delta}$ are calculated as before, and $Z f_{\alpha\beta} f_{\alpha\beta}^T Z)$ can be calculated similarly. After some straightforward calculations we get:

$$\begin{aligned}
\tau(G(2)) &= \Big((1+z_{\delta\delta}-2z_{\gamma\delta}+z_{\delta\delta}) \\
&\quad - \frac{[(z_{\gamma\alpha}-z_{\gamma\beta})-(z_{\delta\alpha}-z_{\delta\beta})]^2}{1+z_{\alpha\alpha}-2z_{\alpha\beta}+z_{\beta\beta}}\Big) \\
&\quad \times (1+z_{\alpha\alpha}-2z_{\alpha\beta}+z_{\beta\beta})\tau(\mathcal{G}_0) \\
&= \Big[(1+R_{eff}(\alpha,\beta))(1+R_{eff}(\gamma,\delta)) \\
&\quad - \left((z_{\gamma\alpha}-z_{\gamma\beta})-(z_{\delta\alpha}-z_{\delta\beta})\right)^2\Big]\tau(\mathcal{G}_0) \qquad (4.36)
\end{aligned}$$

It can be seen that if the term $((z_{\gamma\alpha}-z_{\delta\alpha})-(z_{\delta\alpha}-z_{\delta\beta}))^2$ were absent, the number of spanning tress would increase by a factor of $(1+R_{eff}(\alpha,\beta))(1+R_{eff}(\gamma,\delta))$. In that case it would suffice to join the two pairs of nodes with the highest effective resistance distance to maximize the number of spanning trees. However, this is not true in a general graph due to the interaction term $((z_{\gamma\alpha}-z_{\gamma\beta})-(z_{\delta\alpha}-z_{\delta\beta}))^2$ in equation (4.36). Therefore, adding two edges (α,β) and (γ,δ) with the highest effective resistance distance, will result in the maximum spanning tree only in the symmetric cases where the nodes α and β are situated symmetrically with respect to nodes γ and δ. This is in line with the result of equation (4.32) which requires symmetry with regard to effective resistance distances.

The explicit formula for the cases of adding 3 or more edges can be derived in the same manner by using the Sherman-Morrison-Woodbury formula recursively. As the number of edges increases, more complex terms representing the interaction of the added

edges appear in the formula. It is worthwhile to notice that two factors determine the optimal graph: minimizing a notion of distance (effective resistance) and at the same time symmetrizing the graph. The resulting graph is the result of the interaction and possibly trade-off between these two criteria. Such interaction and trade-off can be observed as the basic phenomenon in the formation of small world graphs, where the base graph provides necessary symmetry, while the shortcuts provide decrease in distance.

In the sequel we first consider the special case of adding a shortcut to a ring, which can already be solved using Theorem 4.8.1. However, solving the problem explicitly provides more insight on the way the edge addition increases the number of spanning trees. We then use the ring graph as a base for a small world network and study the small world effect as far as increase in the number of spanning trees is considered.

4.8.3 Special case: adding a shortcut to a ring

Consider $\mathcal{G}_0$ to be a ring with the corresponding Laplacian matrix $L = D(0) - A(0)$ and natural random walk matrix $P_0 = (D_0)^{-1}(A_0)$. Take an arbitrary node. Without loss of generality we label this node as 1, and label the rest of the nodes as $2, 3, ..., n$ in a clockwise way (Figure 4.9). If $2 < j < n - 1$, we refer to the potential edge $(1, j)$ as a shortcut. The length of such a shortcut is $j - 1$. Let $\mathbb{G} = \{G^{(i)}\}_{i=3}^{n-1}$ denote a set of graphs where G^i denotes the ring with an augmented shortcut between the nodes 1 and i. Denote the corresponding matrices by $\mathbb{L} = \{L^{(i)}\}_{i=3}^{n-1}$ and $\mathbb{P} = \{P^{(i)}\}_{i=3}^{n-1}$.

The number of spanning trees of a ring with n nodes is n. The problem is to find the graph $G^{(k)}$ for which $\tau(G^{(k)})$ is maximized. Since the node degrees are equal in all

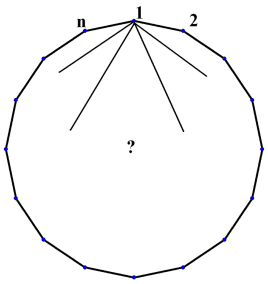


Figure 4.9: Selection of the best shortcut

$G^{(i)}$, using equation (3.28), the term $\prod_{i \neq k} d_i$ is equal for all configurations and it suffices to maximize $\tau_1 = det(Q_1)$.

The corresponding random walk matrix for the ring G_0 is:

$$P^{(0)} = \begin{pmatrix} 0 & \frac{1}{2} & 0 & . & ... & \frac{1}{2} \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 & ... & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} & ... & 0 \\ . & . & . & . & ... & . \\ . & . & . & . & ... & . \\ \frac{1}{2} & 0 & . & ... & \frac{1}{2} & 0 \end{pmatrix} \tag{4.37}$$

Now consider the graph $G^{(i)}$ which is made from G_0 by adding a link between nodes 1 and i and its corresponding matrix $P^{(i)}$. Notice that $P^{(i)}$ only differs from $P^{(0)}$ in the rows 1 and i. The following theorem characterizes the increase in the number of spanning trees as a function of the length of the shortcut.

Theorem 4.8.2. *In the problem of adding a shortcut to a ring, the number of spanning trees is an increasing function of the length of the shortcut. The maximum is attained by the graph $G^{(\frac{n+2}{2})}$ in case n is even. If n is odd the maximal value is attained by graphs*

$G^{(\frac{n+1}{2})}$ *and* $G^{(\frac{n+3}{2})}$.

We need the following Lemma to prove Theorem 4.8.2.

Lemma 4.8.3. *[100] The inverse of the* n *by* n *tridiagonal matrix*

$$B = \begin{pmatrix} b_1 & 1 & 0 & 0 & ... & 0 \\ 1 & b_2 & 1 & 0 & ... & 0 \\ 0 & 1 & b_3 & 1 & ... & 0 \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ 0 & 0 & 0 & ... & 1 & b_n \end{pmatrix} \tag{4.38}$$

with $b_i = b_{n-i+1}$ is an n by n matrix $C = c_{ij}$ with

$$c_{ij} = \begin{cases} r^{-1} r_{i-1} r_{n-i} & i \leq j, \\ c_{ji} & i > j, \end{cases}$$

where $r_0 = 1, r_1 = -b_1, r_k = -(b_k r_{k-1} + r_{k-2}), k = 1, 2, ..., n-1$, and $r = b_n r_{n-1} + r_{n-2}$.

Proof. (Of Theorem 4.8.2) We prove the theorem by directly calculating $\tau_1(G^{(i)}) = det(Q_1^{(i)})$, for $i = 3, ..., n-2$. $Q_1^{(i)}$ is the first principal sub-matrix of $I - P^{(i)}$, that is the matrix which remains after removing the first row and the first column of $I - P^{(i)}$.

The matrices $\{Q_1^{(i)}\}_{i=3}^{n-2}$ are $n-1$ dimensional tridiagonal matrices. Notice that for $Q_1^{(i)}$ the triple on the $(i-1)^{st}$ row is $[-1/3 \quad 1 \quad -1/3]$, which is different from the other row's triplet $[-1/2 \quad 1 \quad -1/2]$. We write this matrix as the sum of two matrices, one which is the "base" tridiagonal matrix, which has equal vectors $[-1/2 \quad 1 \quad -1/2]$ on the diagonal, and is the same for all $Q_1^{(i)}$s. The other matrix is a deviation matrix which is specific to the i^{th} graph.

Let e_i denote the i^{th} coordinate vector, whose only nonzero element is the i^{th} entry which is equal to 1. Then we can write:

$$Q_1^{(i)} = Q_1^{(0)} + \Delta(i) \tag{4.39}$$

In this equation:

$$Q_1^{(0)} = \begin{pmatrix} 1 & -\frac{1}{2} & 0 & 0 & ... & 0 \\ -\frac{1}{2} & 1 & -\frac{1}{2} & 0 & ... & 0 \\ 0 & -\frac{1}{2} & 1 & -\frac{1}{2} & ... & 0 \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ 0 & 0 & 0 & ... & -\frac{1}{2} & 1 \end{pmatrix} \tag{4.40}$$

and

$$\Delta(i) = \frac{1}{6} e_{i-1}(e_{i-2} + e_i)'. \tag{4.41}$$

We now use the identity $det(A + xy') = (1 + y'A^{-1}x)det(A)$ and the fact that $Q_1^{(0)}$ is invertible to calculate $Q_1^{(i)}$.

Let $Z = (Q_1^{(0)})^{-1}$, Then:

$$\begin{aligned} \det(Q_1^{(i)}) &= \det[Q_1^{(0)} + \frac{1}{6} e_{i-1}(e_{i-2} + e_i)'] \\ &= [1 + \frac{1}{6}(e_{i-2} + e_i)' Z e_{i-1}] \det(Q_1^{(0)}) \\ &= \left[1 + \frac{1}{6}(z_{i-2,i-1} + z_{i,i-1})\right] \det(Q_1^{(0)}) \end{aligned} \tag{4.42}$$

Using Lemma 4.8.3 with $B_0 = -2\det(Q_1^{(0)})$,

$$z_{i-2,i-1} = \frac{2}{n}(i-2)(n-i+1),$$

and

$$z_{i,i-1} = \frac{2}{n}(i-1)(n-i).$$

Therefore,

$$\det(Q_1^{(i)}) = \left[1 + \frac{2}{3n}\left((i - \frac{n+2}{2})^2 + \frac{n^2-2n+4}{4}\right)\right]\det(Q_1^{(0)}) \tag{4.43}$$

Equation (4.43) shows that the number of spanning trees is an increasing function of the length of the shortcut. Furthermore the maximum is attained by the graph $G^{(\frac{n+2}{2})}$ in case n is even. If n is odd the maximal value is attained by graphs $G^{(\frac{n+1}{2})}$ and $G^{(\frac{n+3}{2})}$. □

4.9 Small world effect and spanning trees

In this section we consider the increase in the number of spanning trees in the Watts-Strogatz model for the small world effect. We adopt the same framework as in section 4.3 and study the increase of the number of spanning trees in the graph as a result of adding shortcuts with small weights. Consider the base lattice to have a ring topology on n nodes, $\mathcal{G}_0 = C(n,2)$, with adjacency matrix

$$A_0 = \begin{pmatrix} 0 & 1 & 0 & ... & 0 & 1 \\ 1 & 0 & 1 & 0 & ... & 0 \\ 0 & 1 & 0 & 1 & ... & 0 \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ 1 & 0 & 0 & ... & 1 & 0 \end{pmatrix} \tag{4.44}$$

There are n spanning trees in a ring of size n. Instead of shortcuts with small probability, we assume applying negligible weights, ϵ to non-neighboring nodes. The resulting perturbed adjacency matrix is therefore:

$$A_\epsilon = \begin{pmatrix} \epsilon & 1 & \epsilon & ... & \epsilon & 1 \\ 1 & \epsilon & 1 & \epsilon & ... & \epsilon \\ \epsilon & 1 & \epsilon & & ... & \epsilon \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ 1 & \epsilon & \epsilon & ... & 1 & \epsilon \end{pmatrix} \tag{4.45}$$

In the perturbed system, each node's degree is equal to the sum of the weights of the corresponding rows of the adjacency matrix, $2 + (n-2)\epsilon$. Denote $D_\epsilon = (2 + (n-2)\epsilon)I$, the corresponding random walk matrix is equal to:

$$P_\epsilon = D_\epsilon^{-1} A_\epsilon = \frac{1}{2 + (n-2)\epsilon} A_\epsilon. \tag{4.46}$$

The Laplacian and normalized Laplacian matrices (L_ϵ,$\mathcal{L}_\epsilon$) can be defined similarly. Let P_0 given by equation (4.37) denote the random walk matrix corresponding to the unperturbed graph $\mathcal{G}_0$. We can write P_ϵ in terms of P_0:

$$P_\epsilon = \frac{2(1-\epsilon)}{2 + (n-2)\epsilon} P_0 + \frac{\epsilon}{2 + (n-2)\epsilon} J. \tag{4.47}$$

The following lemma determines the eigenvalues of P_ϵ in terms of those of P_0.

Lemma 4.9.1. *The eigenvalues of P_ϵ are*

$$\begin{aligned}\lambda_1(P_\epsilon) &= 1,\\ \lambda_i(P_\epsilon) &= \frac{2(1-\epsilon)}{2+(n-2)\epsilon}\lambda_i(P_0), \quad i = 2,3,...,n.\end{aligned} \tag{4.48}$$

Proof. Consider the matrix

$$P_1 = \frac{2+(n-2)\epsilon}{2(1-\epsilon)}P_\epsilon = P_0 + \frac{\epsilon}{2(1-\epsilon)}J.$$

Then,

$$\begin{aligned}\det(P_1 - \lambda I) &= \det(P_0 - \lambda I + \frac{\epsilon}{2(1-\epsilon)}\mathbf{1}\mathbf{1}^T)\\ &= [1 + \frac{\epsilon}{2(1-\epsilon)}\mathbf{1}^T(P_0 - \lambda I)^{-1}\mathbf{1}]\det(P_0 - \lambda I)\end{aligned} \tag{4.49}$$

Furthermore, for any $\lambda \notin Spec(P_0)$,

$$(P_0 - \lambda I)^{-1}\mathbf{1} = (1-\lambda)^{-1}\mathbf{1},$$

Therefore for such λ,

$$\det(P_1 - \lambda I) = \left(1 + \frac{n\epsilon}{2(1-\epsilon)(1-\lambda)}\right)\det(P_0 - \lambda I) \tag{4.50}$$

It follows that the eigenvalues of P_1 are the same as the eigenvalues of P_0 except for $\lambda_n(P_1) = 1 + \frac{n\epsilon}{2(1-\epsilon)}$.

Since $P_\epsilon = \frac{2(1-\epsilon)}{2+(n-2)\epsilon}P_1$,

$$\begin{aligned}\lambda_1(P_\epsilon) &= 1,\\ \lambda_i(P_\epsilon) &= \frac{2(1-\epsilon)}{2+(n-2)\epsilon}\lambda_i(P_0), \quad i = 2,3,...,n-1,\end{aligned}$$

and the result follows. □

Now, we can state the following proposition:

Proposition 4.9.2. *Let $\epsilon = \frac{K}{n^\alpha}$, $\alpha > 1$, and consider the ratio $r = \frac{\tau(\mathcal{G}_\epsilon)}{\tau(\mathcal{G}_0)}$, which measures the increase in the number of spanning trees as a result of adding ϵ weights:*

- *For $\alpha > 3$, the effect of perturbation on is negligible.*
- *$\alpha = 3$ is the onset of the effectiveness of shortcuts.*
- *For $1 < \alpha \leq 3$, the shortcuts dominantly increase the number of spanning trees, i.e. $\lim_{n\to\infty} \frac{\tau(\mathcal{G}_\epsilon)}{\tau(\mathcal{G}_0)} = \infty$.*

Proof. Using equation (3.28),

$$r = \frac{\tau(\mathcal{G}_\epsilon)}{\tau(\mathcal{G}_0)} = \left(\frac{\prod_{j=2}^{n}(1-\lambda_j(P_\epsilon))}{\prod_{j=2}^{n}(1-\lambda_j(P_0))}\right) \cdot \left(\frac{\prod_{i=1}^{n} d_i(\mathcal{G}_\epsilon)}{\prod_{i=1}^{n} d_i(\mathcal{G}_0)}\right) \cdot \left(\frac{\sum_{i=1}^{n} d_i(\mathcal{G}_0)}{\sum_{i=1}^{n} d_i(\mathcal{G}_\epsilon)}\right) \tag{4.51}$$

The product consists of three terms. We first show that the second and third terms are constant for the range of α that we are interested in.

- The second term:

$$\frac{\prod_{i=1}^{n} d_i(\mathcal{G}_\epsilon)}{\prod_{i=1}^{n} d_i(\mathcal{G}_0)} = [\frac{2+(n-2)\epsilon}{2}]^{n-1} = [1 + \frac{K}{2n^{\alpha-1}} - \frac{K}{n^\alpha}]^{n-1}$$

  For $\alpha > 2$ the limit of this term for large n is 1. For $\alpha = 2$ the limit is constant.

- The third term:

$$\frac{\sum_{i=1}^{n} d_i(\mathcal{G}_0)}{\sum_{i=1}^{n} d_i(\mathcal{G}_\epsilon)} = \frac{2n}{(2+(n-2)\epsilon)n} = \frac{2n}{2n + Kn^{2-\alpha} - 2Kn^{1-\alpha}}.$$

  For $\alpha > 1$ the limit for large n is 1.

Therefore, the limit for $2 < \alpha$ is solely determined by the first term. We notice that the matrix P_0 is a circulant matrix with $\lambda_1 = 1$. For even n the rest of the eigenvalues have multiplicity 2 and come in pairs:

$$\lambda_i = \cos\frac{2\pi i}{n}, \quad i = 2, 3, ..., \frac{n}{2}.$$

For odd n, we have

$$\lambda_i = \cos\frac{2\pi i}{n}, \quad i = 2, 3, ..., \frac{n+1}{2},$$

with multiplicity 2, and an additional single eigenvalue $\lambda_{\frac{n+1}{2}}$.

To evaluate the first term, note that equation (4.48) indicates that $\lambda_i(P_\epsilon) < \lambda_i(P)$ for $i = 2, 3, ..., n$. A straightforward calculation also shows that$\frac{1-\lambda_i(P_\epsilon)}{1-\lambda_i(P_0)}$ is a decreasing function of i for $i = 2, 3, ..., \lfloor\frac{n+1}{2}\rfloor$, with λ_2 attaining the maximum ratio. Therefore, we can write:

$$\left(\frac{1-\lambda_2(P_\epsilon)}{1-\lambda_2(P_0)}\right)^2 \leq \left(\frac{\prod_{j=2}^{n}(1-\lambda_j(P_\epsilon))}{\prod_{j=2}^{n}(1-\lambda_j(P_0))}\right) \leq \left(\frac{1-\lambda_2(P_\epsilon)}{1-\lambda_2(P_0)}\right)^{\lfloor\frac{n-1}{2}\rfloor} \tag{4.52}$$

We now consider the ratio $\frac{1-\lambda_2(P_\epsilon)}{1-\lambda_2(P_0)}$ in the limit of large n. We approximate $\cos\frac{2\pi}{n}$ by its first two terms of Taylor series expansion, $1 + \frac{2\pi^2}{n^2}$. In the limit of large n,

$$\frac{1-\lambda_{n-1}(P_\epsilon)}{1-\lambda_{n-1}(P_0)} = \frac{1-\frac{2(1-\epsilon)}{2+(n-2)\epsilon}.(1-2\pi^2 n^{-2}+\text{h.o.t})}{2\pi^2 n^{-2}+\text{h.o.t},}$$

in which h.o.t denotes higher order terms. Replacing $\epsilon = Kn^{-\alpha}$ and further simplification yields:

$$\frac{1-\lambda_{n-1}(P_\epsilon)}{1-\lambda_{n-1}(P_0)} = \frac{Kn^{1-\alpha}+4\pi^2 n^{-2}+\text{h.o.t}}{4\pi^2 n^{-2}+2K\pi^2 n^{-1-\alpha}+\text{h.o.t}} \tag{4.53}$$

For $\alpha < 3$, this ratio is 1. At $\alpha = 3$, the inequalities (4.52) become nontrivial. As we further decrease $\alpha < 3$, $\frac{1-\lambda_2(P_\epsilon)}{1-\lambda_2(P_0)} \to \infty$ in the limit of large n. Therefore $\alpha = 3$ is the onset of the small world effect. □

As we have argued in section 4.5, this probabilistic interpretation leads to a construction of small world networks by switching between graphs with low probability shortcuts. At each switching interval a few shortcuts are generated uniformly. Similarly, one can think of generating random spanning trees. In a recent paper Goyal et al. [42] have shown that the union of a few random spanning trees has constant edge expansion ratio and can be considered as *expander graphs.*

Efficient heuristics should consider symmetrizing the graph and adding edges between nodes with high resistance distance. Since the problem of adding one or two shortcuts is less complex, an efficient method is to solve such smaller problem, determine the best choices of edge augmentation for a set of nodes, and probabilistically switch between these configurations.

Chapter 5

Invariant nodes and a hierarchical scheme for fast convergence of distributed algorithms

5.1 A hierarchical self organizing method

In the previous sections, we showed the effect of graph topology on the convergence speed of consensus problems and used consensus algorithms to discriminate between graph topologies. We showed that small world phenomenon results as nodes placed on a suitable base graph use $\epsilon-$shortcuts, which is equivalent to a random walk using shortcuts with small probabilities. In a broader view, this can be seen as local connectivity provided by some grid or other clustered local graph and global connectivity provided by shortcuts. In social systems such a phenomenon can be observed in a society where the communities represent local connectivity and the intercommunity links correspond to shortcuts. The problem of finding communities in networks has been considered in the context of complex networks [80], and its possible relation to small world phenomenon has been pointed out [24]. Extracting efficient community structures in graphs is useful for example, in backbone construction in wireless ad hoc networks [70].

In this section we show that a very efficient communication pattern with substantial improvement in performance is possible by a two level hierarchical scheme. The idea here is that selecting a few well connected and controlled agents which are well pro-

tected should enhance the speed of convergence of distributed schemes like consensus algorithms.

Given n agents, suppose we can divide them into K groups each having M members, so $n = K.M$. It is assumed that $K \leq M \ll n$. The exact sizes of K and M are problem specific and influence the performance of the algorithm. For each group suppose that we can select a "leader". The leaders should be able to have two properties: they should be well connected to the members of their group, and they should also be able to communicate with other leaders when necessary. If the distributed algorithm is carried out at each group separately and the leaders communicate on a higher level, the agents can enjoy faster convergence rates; the reduction of the size of each group from n to $M \leq \sqrt{n}$ results in faster intergroup convergence whereas the ease of communication between the leaders upon demand results in overall fast convergence.

We now provide a semi-distributed method which can categorize the agents as "leader" or "regular". Further, the method assigns each regular agent with an influence vector which indicates which leader has more influence on it. This provides the nodes with some global picture of the network.

5.2 Distributed exploration of the graph structure

The structure of a graph plays a crucial role in properties of a distributed algorithm that is running on it. Given a graph, individual nodes have only local knowledge about its structure, which includes information about their neighboring nodes. If any node wants to either improve its own performance or a global performance measure it needs to know

more about the global picture of the network. This information can be used by the node to refine its choice of neighbors in order to improve its performance.

The most complete measure of global graph structure is the adjacency matrix. Since each node has limited memory, energy, and computational capacity, they cannot store and process the adjacency matrix. Our goal is to devise a scheme to provide each node with a small vector that includes compact global information on how the node is located with respect to the other nodes. It is desired that the scheme can be disseminated via an implementable distributed manner.

We propose a two stage algorithm for this purpose. Apart from a single data transmission and reception at a central authority by each node, the algorithm is carried out in a decentralized manner. In the first stage nodes will collaborate to find their *social degree* [13]. This is a local measure of how 'well connected' each node is. Once the nodes find out their social degree, they will transmit it to a central authority which determines K "social leaders" of the graph- the better connected nodes among all. The central authority then broadcasts the list of K social leaders to all of the nodes. In the second stage, each node uses a simple iterative scheme to maintain its *influence vector*, a vector of size K, which determines the influence of each social leader on it. We acknowledge that while developed independently, the idea has similarities to recent community evaluation schemes developed in network science literature [80].

5.3 Social degrees and leader nodes

To find the leaders or the agents with the highest influence we use a generalization of a framework proposed by Blondel *et al* [13]. They define the social degree of a node as the number of the cycles of length 3 passing through that node. They also define a social leader as a node with the highest degree in its neighborhood. This can be generalized as:

Definition 2. Social degree of order k *of a node (denoted by* $SD^{(k)}(v_i)$*) is defined to be the number of cycles of length* k *passing through the node, if* $k > 2$ *and the number of its neighbors, if* $k = 2$*.*

Definition 3. *A* leader node of order k *is the node with the highest social degree of order* k *among its neighbors.*

Definition 4. *For given* $0 < \alpha < 1$ *and* $\beta = 1 - \alpha$*, a node's* social scores *is defined as* $SC(v_i) = \alpha.SD^{(2)}(V_i) + \beta.SD^{(3)}(v_i)$.

Notice that each node can determine its social degree of orders 2 and 3 by a simple query from its neighbors. Since determining higher order degree requires more effort, we use the orders 2 and 3 for our present application.

In the first stage of the algorithm, each node computes its social degrees of order 2 and 3. It also queries the social degrees of its neighbors. Upon comparing its social degrees with its neighbors, if a node is found to be a leader of order 2 or 3, it transmits its degrees to the central authority. Upon receiving these data from the leader nodes, the central authority selects K nodes $l_i, i = 1, ..., k$ with the highest social scores, $SC(l_i)$, and gives an arbitrary order to them and transmits their assigned order to them. Notice

that the choice of α and β determines the preference between leaders in a "star-like" neighborhood versus leaders of better connected neighborhoods. Once a selected leader is assigned its order $1 \leq i \leq K$ it will maintain the constant vector $e_i \in R^K$. This is the unit vector with 1 in its ith entry.

5.4 Determination of the influence vector

Our objective in this part is to associate with each of the regular nodes a vector that determines how well it is related to each of the leaders and how it is influenced by them. The amount of influence that a leader has on a local node is not only determined by their distance but also by the number of paths between them. We provide a definition for the influence vector based on the properties of random walks on graphs.

Definition 5. *Consider a graph with K leaders and $n - K$ regular nodes. The influence of leader nodes l_k ($k = 1, ..., K$) on any regular node i is the probability that a random walk that starts from i hits l_k before it hits any other leader node.*

Given the leaders and the arbitrary order assigned to them, we first describe the algorithm to determine the influence vectors for each regular node. Then we will show why it converges and why it outputs valid vectors as influence vectors. We denote the influence vector of node i by $x_i \in R^K$. By $x_i^k(t)$ we mean the kth entry of the influence vector of node i evaluated at time t.

The algorithm operates as follows. The influence vector of leader l_i is first assigned to be the unit vector $x_i = e_i$. These K vectors do not vary. For all regular nodes i, x_i is initialized randomly, distributed uniformly on $[0, 1]^K$. At each iteration time $t + 1$,

whereas each regular node i, updates its influence vector entry-wise ($k = 1, 2, ..., K$) using the following rule.

$$x_i^k(t+1) = \frac{1}{1+n_i(t)}[x_i^k(t) + \sum_{j \in N_i(t)} x_j^k(t)] \tag{5.1}$$

Theorem 5.4.1 shows the effectiveness of the above scheme.

Theorem 5.4.1. *If the underlying graph is connected, the iteration (5.1) converges to a set of unique vectors. Furthermore, $lim_{t\to\infty} x_i^k(t)$ is equal to the probability that a random walk starting at node i hits the leader node l_k before any other leader node.*

Proof. The proof follows from Theorem 2.5.2 The particular form of the solutions arises because the procedure solves a discrete version of the Dirichlet problem on the graph. We follow the proof of Bremaud [18]. Relabel the nodes, such that $D = \{1, 2, ..., n-K\}$ denote the regular nodes and $\partial D = \{n-K+1, ..., n\}$ denote the leader nodes, where $l_i = n - K + i$. For all $k = l_1, ..., l_K$, define a function ϕ^i on the graph such that $\phi^k(l_j) = \delta(k, l_j)$, where δ is the Dirac Delta function.

Let $P = (I + D)^{-1}(A + D)$, where A is the adjacency matrix of the graph G and D is the diagonal matrix with i th diagonal element equal to the degree of node i. Fix k, i.e. consider the influence of leader l_k. Then x_i^k converges to a value that satisfies the following equation

$$x_i^k = \begin{cases} (Px^k)_i & i = 1, 2, ..., n-K \\ \phi^k(i) & i = n-K+1, ..., n \end{cases} \tag{5.2}$$

Note that P is a stochastic matrix and the first equation is valid for the regular nodes. Let $\{Z_n^k\}_{n\geq 0}$ be a homogeneous Markov chain with state space $V = \{1, 2, ..., n\}$.

Let T be the hitting time of ∂D. For each state $i \in V$ define:

$$h_i^k = E[\phi(Z_T^k)|Z_0^k = i] \tag{5.3}$$

Since the underlying graph G is connected, P is irreducible. Also $\forall i \in V$, $p_{ii} > 0$, which means the chain is aperiodic. The number of states is finite and therefore the chain is positive recurrent and $P(T < \infty|Z_0^k = i) = 1$.

By definition $h^k = \phi^k$ on ∂D and $x^k = \phi^k$ on D. By first step analysis: $h_i^k = \sum_{j \in V} p_{ij} h_j^k$ on D. So:

$$h^k = \begin{cases} Px^k & on \quad D \\ \phi^k & on \quad \partial D \end{cases} \tag{5.4}$$

Therefore $h = x$ on the graph G. The proof of uniqueness of the solutions also follows from [18]. Notice that ϕ_i^k is defined such that h_i^k is equal to the probability of hitting the leader node l_k before the other leader nodes. □

Chapter 6

Sensor scheduling using smart sensors

6.1 Sensor scheduling

Recently there has been a lot of interest in networks of sensing agents which act cooperatively to obtain the best estimate possible, e.g., see [50, 98] and the references therein. While such a scheme admittedly has higher complexity than the strategy of treating each sensor independently, the increased accuracy often makes it worthwhile. If measurements from all the sensors are pooled, the resulting estimate can be even better than the one based on the sensor with the least measurement noise (where no information exchange occurs).

Communication constraints, however, often impose a restriction on the maximum number of sensors that can transmit data to the estimator. Thus, there is a problem of sensor scheduling. One example when such a situation arises is when there are echo-based sensors like sonars which can interfere with each other. Another situation where sensor scheduling is useful is in tracking and discrimination problems, where a radar can make different types of measurements by transmitting suitable waveforms, each of which has a different power requirement. There might be shared communication resources (e.g., broadcast channels or a shared communication bus) that constrain the usage of many sensors at the same time. Such a situation arises, e.g., in telemetry-data aerospace systems.

Because of its importance, the sensor scheduling problem has received considerable

attention in the literature. The seminal work in [74] proved a separation property between the optimal plant control policy and the measurement control policy for LQ control. The measurement control problem, which is the sensor scheduling problem, was cast as a non-linear deterministic control problem and shown to be solvable by a tree-search in general. It was proven that if the decision to choose a particular sensor rests with the estimator, an open-loop selection strategy is optimal for a cost based on the estimate error covariance. Forward dynamic programming and a gradient method were proposed for this purpose. To deal with the complexity of a tree-search, greedy algorithms have been proposed many times, some examples being [61, 87]. Allied contributions have dealt with robust sensor scheduling [99], a greedy algorithm with an information based cost measure [125] and the works of [69, 76, 94]. A different numerical approach to solve the problem was provided in [2] which cast the problem as a two-point boundary value problem. This approach was further considered in [54, 67]. A completely general framework for nonlinear systems and general nonlinear diffusion sensor signals was developed in the seminal paper [8]. The dynamic sensor scheduling problem was solved using dynamic programming methods, based on general stochastic control separation and nonlinear filtering, which involved quasi-variational inequality techniques for the analytical proofs [8]. A stochastic algorithm that is particularly useful in situations where communication channels impose random data dropouts was proposed by Gupta *et al.* in [44].

However, these approaches assume that a sensor, when allowed to transmit at time step k, transmits only the latest measurement that it observed at time step k. Thus, even if all sensors are taking measurements at every time step, the estimator does not have access to all this information. A notable exception is the general framework and meth-

ods of [8], where the estimator has complete past histories of measurements, and where even simultaneous measurements by several sensors in each time step are allowed. In networked control systems, sensors are usually equipped to communicate over wireless channels or communication networks. Thus, it is reasonable to assume that they possess some storage and processing capabilities. Thus, if the sensors can execute simple recursive algorithms to process the information being collected, significant improvement in estimation (or control) performance can be expected. Such algorithms have already been considered by Gupta and coauthors and demonstrated for the case of single sensor systems in [46, 45]. In [49], the improvement in the stability region using such pre-processing strategies for multi-sensor systems is illustrated. Here, we use information processing algorithms along the lines of the ones proposed in [48] for the sensor scheduling problem. As we shall see, the optimal algorithms for the sensor scheduling problem require much less data communication than the general multi-sensor problem, since only one sensor transmits at every time step.

Using these information processing algorithms, we show that we obtain significantly better estimates. We also consider the problem of finding the optimal sensor schedule. While the general solution remains a tree-search, we show that the number of paths to be searched are significantly pruned. We also prove a periodicity result in the optimal sensor schedules.

This chapter is organized as follows. The next section deals with the problem formulation. The recursive optimal information processing algorithm of [48] is presented in section 6.3. In Section 6.4, we consider the problem of optimal scheduling. Finally, in Section 6.5, we study the case when the decision (selection) is between a sensor transmit-

ting or not, and present a periodicity result, which also applies to more general scenarios.

6.2 Modelling and Problem Formulation

Consider a system evolving as

$$x(k+1) = Ax(k) + w(k), \tag{6.1}$$

where $x(k) \in \mathbf{R}^n$ is the process state at time step k and $w(k)$ is the process noise assumed white, Gaussian and zero mean with covariance matrix R_w. The initial condition $x(0)$ is assumed independent of the process noise and Gaussian with zero mean and covariance P_0. The process state is being observed by N sensors S_1, S_2, $\cdots$, S_N with the measurement equation for the i-th sensor being

$$y_i(k) = C_i x(k) + v_i(k), \tag{6.2}$$

where $y_i(k) \in \mathbf{R}^{s_i}$ is the measurement. The measurement noises $\{v_i(k),\ i = 1,\ \cdots,\ N\}$, for the sensors are assumed independent of each other, of the process noise and of the initial condition. Further the noise $v_i(k)$ is assumed to be white, Gaussian and zero mean with covariance matrix R_i. We will assume $N = 2$. The ideas are applicable to the general case, at the expense of more notation and communication. We assume that the pair (A, C) is observable and the pair $(A, R_w^{\frac{1}{2}})$ is stabilizable, where $C = \begin{bmatrix} C_1^T & C_2^T \end{bmatrix}^T$.

At every time step k, one sensor is chosen to take the measurement. If the i-th sensor is chosen at time k, we represent this event as $t(k) = i$. By a sensor schedule, we mean the choice of events $t(0), t(1), \cdots$. The i-th sensor then calculates a finite vector

$$s_i(k) = f(i, k, y_i(0), \cdots, y_i(k), t_i(0), \cdots, t_i(k)),$$

where $s_i(k) \in \mathbf{R}^m$ and transmits it to a central estimator (equivalently, shared with all the sensors) in an error-free manner. By abusing the notation a bit, we denote by $s(k)$ the vector received by the estimator at time step k. The estimator calculates an estimate

$$\hat{x}(k+1) = g\left(k, s(0), s(1), \cdots, s(k)\right)$$

of the state $x(k+1)$ that minimizes the usual mean squared error

$$P(k+1) = E\left[e(k)e^T(k)\right]$$

where $e(k)$ is the error defined as

$$e(k) = x(k+1) - \hat{x}(k+1).$$

We can compare the performance of particular encoding functions $f()$ and decoding functions $g()$ by comparing the finite-horizon cost

$$J_K = \sum_{k=1}^{K} trace(P(k)),$$

or the infinite-horizon cost

$$J_\infty = \lim_{K\to\infty} \frac{1}{K} \sum_{k=1}^{K} trace(P(k)).$$

We are concerned with the following problems:

1. What are the functions f and g that are optimal with respect to the cost function J for any schedule of the sensors?

2. What is the optimal sensor schedule for the infinite-horizon cost? We will be interested in open loop schedules where the choice of the event $t(k)$ does not depend on the measurement values $\{y_i(k),\ i\ =\ 1,\ \cdots,\ N\}$.

3. For the special case when the sensing choices consist of transmitting a measurement by the sensor or not transmitting one, what is the optimal schedule for transmitting measurements for the finite-horizon cost?

We begin in the next section by solving for the optimal encoding and decoding functions.

6.3 Optimal Encoding and Decoding Functions

At any time k, define the time-stamp corresponding to sensor i as

$$\tau_i(k) = \max\{j \mid j \leq k, \quad t(j) = i\}.$$

Thus the time-stamp denotes the latest time at which transmission was possible from sensor i. Using the time-stamp, we can define the maximal information set $\mathcal{I}_i^{\max}(k)$ for each sensor as

$$\mathcal{I}_i^{\max}(k) = \{y_i(0), y_i(1), \cdots, y_i(\tau_i(k))\}.$$

The maximal information set is the largest set of measurements from sensor i that the controller can possibly have access to at time k. For any encoding functions f chosen by the sensors, the information available at the estimator will be a sub-set of the maximal information set. Hence, with the optimal minimum mean squared error (MMSE) estimation being chosen as the decoding function g by the decoder, the performance for any encoding functions f will be upper bounded (equivalently, the cost will be lower bounded) if the estimator had access to the maximal information sets from all the sensors.

Now consider an algorithm $\bar{\mathcal{A}}$ under which at every time step k, if $t(k) = i$, every

sensor i transmits the set

$$S_i(k) = \{y_i(0), y_i(1), \cdots, y_i(k)\}.$$

Note that the algorithm $\bar{\mathcal{A}}$ does not specify valid encoding functions since the dimension of the transmitted vectors cannot be bounded by any constant m. However, if the algorithm $\bar{\mathcal{A}}$ is followed, at any time step k, the decoder (and the controller) would have access to the maximal information sets $\mathcal{I}_i^{\max}(k)$. This implies that for any other encoding algorithm, the cost will always be higher for any given schedule than obtained by using the algorithm $\bar{\mathcal{A}}$. Thus, in particular, one way to achieve the optimal value of the cost J_K or J_∞ for a given schedule is through the combination of an encoding algorithm that makes the information sets $\mathcal{I}_i^{\max}(k)$ available to the controller and a controller that optimally utilizes the information set. Further, one such information processing algorithm is the algorithm $\bar{\mathcal{A}}$ described above. However, this algorithm requires increasing data transmission as time evolves. Surprisingly, in a lot of cases, we can achieve the same performance using a constant amount of transmission and memory.

To this end, we begin with a result of [49]and [48]. This result identifies the optimal information processing to be done by the sensors to ensure that the estimator can calculate the estimate of state $x(k+1)$ based on the maximal information sets $\mathcal{I}_i^{\max}(k)$.

Proposition 6.3.1. *Consider a process of the form (6.1) being observed by two sensors of the form (6.2). The estimate $\hat{x}(k|l, m)$ of the state based on measurements from sensor 1 till time l and sensor 2 till time m can be calculated using the algorithm given below. Assume, without loss of generality, that $l \leq m$.*

- *At each time step $j \leq k$, the sensor 1 executes the following actions:*

1. *Let $\hat{x}_i(k|l)$ denote the MMSE estimate of $x(k)$ based on all the measurements of sensor i up to time l. Denote the corresponding error covariance by $P_i(k|l)$. Obtain the estimate $\hat{x}_1(j|j)$ and $P_1(j|j)$ through a Kalman filter. For $j \leq l$, use the measurement $y_1(j)$. For $j > l$, assume that the sensor 1 did not take any measurement at time step j.*

2. *Calculate*

$$\lambda_1(j) = (P_1(j|j))^{-1} \hat{x}_1(j|j) - (P_1(j|j-1))^{-1} \hat{x}_1(j|j-1).$$

3. *Calculate global error covariance matrices $P(j|j,j)$ and $P(j|j-1,j-1)$ using the relation*

$$(P(j|j,j))^{-1} = \begin{cases} (P(j|j-1,j-1))^{-1} + C_1^T (\Sigma_{v,1})^{-1} C_1 \\ +C_2^T (\Sigma_{v,2})^{-1} C_2 & \text{if } j \leq l \\ (P(j|j-1,j-1))^{-1} \\ +C_2^T (\Sigma_{v,2})^{-1} C_2 & \text{if } l < j \leq m \\ (P(j|j-1,j-1))^{-1} & \text{otherwise,} \end{cases}$$

$$P(j|j-1,j-1) = AP(j-1|j-1,j-1)A^T + \Sigma_w.$$

4. *Obtain*

$$\gamma(j) = (P(j|j-1,j-1))^{-1} AP(j-1|j-1,j-1).$$

5. *Finally calculate*

$$I_{1,l,m}(j) = \lambda_1(j) + \gamma(j) I_{1,l,m}(j-1), \tag{6.3}$$

with $I_{1,l,m}(-1) = 0$.

- *The quantity $I_{2,l,m}(k)$ is calculated by a similar algorithm except using the local estimates $\hat{x}_2(j|j)$ and covariance $P_2(j|j)$.*

- *Finally, the estimate $\hat{x}(k|l,m)$ is calculated using the relation*

$$(P(k|k,k))^{-1}\,\hat{x}(k|l,m) = I_{1,l,m}(k) + I_{2,l,m}(k), \tag{6.4}$$

where $P(k|k,k)$ is calculated as above.

Proof. That $\hat{x}(k|l,m)$ is indeed the MMSE estimate given all the measurements from sensor 1 till time l and from sensor 2 till time m can be proved by utilizing the block diagonal structure of the matrix Σ_v as in the proof of Theorem 2 in [46]. □

The above result identifies the quantities that need to be transmitted by the two sensors to calculate the MMSE estimate of $x(k)$. The quantities depend only on local measurements at the sensors; however, an implicit assumption is that each sensor is informed about the times l and m.

We now present an algorithm according to which the sensors can calculate these optimal vectors with constant memory and processing for any given schedule. We present the algorithm $\mathcal{A}_1$ that the 1st sensor needs to implement. The algorithm $\mathcal{A}_2$ for the second sensor is similar.

Algorithm $\mathcal{A}_1$ to be followed by sensor 1: The sensor maintains two vectors $I^1_{1,k,\alpha_2(k)}(k)$ and $I^2_{1,k,k}(k)$.

1. *Initialization:* Initialize both the vectors to be zero vectors.

$$\begin{aligned} I^1_{1,-1,\alpha_2(-1)}(-1) &= 0 \\ I^2_{1,-1,-1}(-1) &= 0. \end{aligned}$$

2. *Update and Transmission:* At every time step $k \geq 0$, there are two cases:

   - Sensor 1 transmits at time step k: It takes the following actions:
     - It updates vector $I^1_{1,k-1,\alpha_2(k-1)}(k-1)$ to calculate $I^1_{1,k,\alpha_2(k)}(k)$ using an algorithm of the form mentioned in Proposition 6.3.1, where $\alpha_2(k) = \alpha_2(k-1)$. It then transmits this vector.
     - It updates the vector $I^2_{1,k,k}(k)$ from $I^2_{1,k-1,k-1}(k-1)$ using an algorithm of the form mentioned in Proposition 6.3.1.
   - Sensor 2 transmits at time step k: Sensor 1 takes the following actions:
     - It updates the vector $I^2_{1,k,k}(k)$ from $I^2_{1,k-1,k-1}(k-1)$ using an algorithm of the form mentioned in Proposition 6.3.1.
     - It resets $I^1_{1,k,\alpha_2(k)}(k) = I^2_{1,k,k}(k)$.

For this algorithm, it can be verified that

1. The index $\alpha_2(k)$ is always equal to the last time $m \leq k$ where sensor 2 was able to transmit.

2. All the update steps at time k require only the knowledge of the latest measurement from sensor 1 $y_1(k)$. Thus, the sensor requires constant memory and processing.

These two observations allow us to state the following result.

Proposition 6.3.2. *Consider the problem formulation stated in Section 6.2. Using the transmitted vectors $I^1_{1,k,\alpha_2(k)}(k)$ and $I^2_{2,\alpha_1(l),l}(l)$ from the two sensors, the estimator can construct the MMSE estimate of $x(k+1)$ using all the measurements from sensor 1 till*

time k and from sensor 2 till time l. Further, the vectors can be calculated by the sensors using constant amount of processing, memory and transmission at every time step using algorithms $\mathcal{A}_1$ and $\mathcal{A}_2$.

Remark 6.3.1. *The algorithm we have outlined is optimal among all other causal encoding algorithms, in the sense that for any given schedule of transmission, the cost J_K achieved at any time K is minimum for this algorithm. It can also be extended to consider the effect of stochastic packet drops by communication channels from the sensors to the estimator. However, we do not proceed in this direction.*

Having identified an algorithm that allows the estimator to calculate the estimate based on all previous measurements from a sensor till its time stamp, we now proceed to the question of identifying an optimal schedule.

6.4 Optimal Scheduling

In this section, we look at designing an optimal schedule, i.e., the choice of the events $t(k)$ at every time step k. We begin by considering the finite horizon cost J_K. We first note that for the optimal encoding and decoding functions that we have identified in Section 6.2, the proof of optimality of open loop schedules [74] can directly be carried over. In other words, the optimal open loop schedule, in which the choice of $t(k)$ depends only on the system parameters, yields the same performance as the optimal closed loop schedule, in which $t(k)$ can additionally depend on the choice of events $t(0)$, $t(1)$, $\cdots$, $t(k-1)$. Thus, from now on, we will consider obtaining the optimal open loop schedule.

All the possible sensor schedule choices can be represented by a tree structure. The

depth of any node in the tree represents time instants with the root representing time zero. The branches correspond to choosing a particular sensor to be active at that time instant. Each node is associated with the cost function evaluated using the sensor schedule corresponding to the path from the root to that node. Obviously, finding *the* optimal sequence requires traversing all the paths from the root to the leaves in the tree. If the leaves are at a depth d, a total of 2^d schedules need to be compared. This procedure might place too high a demand on the computational and memory resources of the system. We will now see that with the optimal encoding and decoding functions, we can prune the tree significantly. This allows us to traverse the tree for a longer time horizon K.

Consider the case when the estimation error covariance, when $x(k+1)$ is estimated using the measurements of *both* the sensors till time step k, has reached a steady state value $P^\star$. The steady-state value exists because of our observability assumptions. Further, the steady-state value is reached exponentially [62]. For simplicity, we will assume that the horizon K is long enough so that the cost incurred in the transient phase is small and can be ignored during the optimization[1]. Thus, we can carry out the optimization by assuming that the steady-state has been reached.

We define the following Riccati operator:

$$h_i(P) = APA^T + R_w - APC_i^T \left(C_i P C_i^T + R_i\right)^{-1} C_i P A^T, \qquad i = 1, 2. \tag{6.5}$$

The operator acts on a positive semi-definite matrix P and results in a value that equals the estimate error covariance at time step $k+1$ assuming that sensor i was used at time step k and the initial error covariance at time step k was P. We also define another operator

[1]Equivalently, we can assume that the covariance of the initial state $P(0) = P^\star$.

that consists of applying the above operator multiple times. We denote

$$h_i^t(P) = \underbrace{h_i(h_i(...(h_i(P))))}_{\text{t times}}, \qquad i = 1, 2, \tag{6.6}$$

in which h_i has been applied t times. We note that

1. $h_i^1(P) = h_i(P)$.

2. $h_i^t(P)$ is an increasing function in the index t for any positive semi-definite matrix P.

The key observation that allows us to prune the tree is the following. When the optimal encoding and decoding functions are employed by the sensors, the effect on the error covariance at the estimator is the same as if all previous measurements were also transmitted by each sensor whenever it was allowed to transmit. That is, if $t(k) = i$, the i-th sensor could be considered to be transmitting all measurements $y_i(0), y_i(1), \cdots, y_i(k)$. Thus, in the steady state, the error covariance at the estimator resets to $h_i(P^\star)$ whenever a switching from sensor j to sensor i happens. Moreover, if no further switching happens in an interval of length t the error covariance at the end of this interval will be $h_i^t(P^\star)$.

This observation allows us to discard many sequences in the search tree and prune it significantly. We have the following result.

Proposition 6.4.1. *Consider the problem formulation stated in Section 6.2. Suppose that the optimal encoding and decoding functions, as identified in Section 6.4 are being followed. Further, assume that the steady-state has been reached, so that the error covariance in estimating the state $x(m+1)$ based on all the measurements from both the sensors*

till time m is $P^\star$. Let the sensors be denoted by i and j. Suppose there exists $k > 0$ such that

- *For $m = 1, ..., k-1$, $Trace(h_i^m(P^\star)) \leq Trace(h_j(P^\star))$*
- $Trace(h_i^k(P^\star) > Trace(h_j(P^\star))$

Define two sub-sequences for selecting the sensors

$$\begin{aligned} S_1 &= \{t(n) = i, t(n+1) = i, \cdots, t(n+k-1) = i\} \\ S_2 &= \{t(m) = j, t(m+1) = j\}, \end{aligned}$$

for arbitrary times m and n. Then, the sub-sequences S_1 and S_2 can not appear in the optimal schedule.

Proof. We will prove that an optimal schedule cannot contain sub-sequence S_1 by contradiction, by showing that the cost incurred by the optimal schedule can be reduced by choosing another sequence if the optimal sequence indeed contains S_1. Denoting the optimal sequence choices by $t^\star(l)$, we assume that the optimal schedule $S^\star$ contains the sequence S_1, such that for some time n, $t^\star(n) = i$, $t^\star(n+1) = i$, $\cdots$, $t^\star(n+k-1) = i$. We can divide the event space into two possibilities:

1. There is at least one time $m \geq n+k$, such that $t^\star(m) = j$. Let τ denote the smallest such time after $n + k$ when sensor j is used. Now consider an alternate schedule S in which the choices are denoted by $t(l)$. The schedule S is constructed using the

choices:

$$t(l) = \begin{cases} t^\star(l) & l \leq \tau - 3 \\ j & l = \tau - 2 \\ i & l = \tau - 1 \\ t^\star(l) & l \geq \tau. \end{cases}$$

The cost achieved using schedule S is less than the cost achieved using schedule $S^\star$. This because the cost incurred at time steps $l \leq \tau - 3$ and $l \geq \tau$ is identical for the two schedules. However, the cost for schedule $S^\star$ at time steps $\tau - 2$ and $\tau - 1$ is $trace(h_i^k(P^\star) + h_j(P^\star))$, while for the schedule S, it is $trace(h_i(P^\star) + h_j(P^\star))$. Since $trace(h_i^k(P^\star)) > trace(h_i(P^\star))$, our assumption is wrong and $S^\star$ being the optimal schedule means that it cannot contain S_1.

2. The other possibility is that for all future time steps $m \geq n + k$ till time K, sensor i is used. However, in that case, we can consider an alternate schedule S in which the choices are denoted by $t(l)$. The schedule S is constructed using the choices:

$$t(l) = \begin{cases} t^\star(l) & l \leq n + k - 2 \\ j & l = n + k - 1 \\ t^\star(l) & l \geq n + k. \end{cases}$$

Once again, the cost achieved using schedule S is less than the cost achieved using schedule $S^\star$. This is because the cost incurred at time steps $l \leq n + k - 2$ and $l \geq n + k$ is identical for the two schedules. However, the cost for schedule $S^\star$ at time step $n + k - 1$ is $trace(h_i^k(P^\star))$, while for the schedule S, it is $trace(h_j(P^\star))$.

Since $trace(h_i^k(P^\star)) > trace(h_j(P^\star))$, our assumption is wrong and $S^\star$ being the optimal schedule means that it cannot contain S_1.

By a similar argument, we can prove that the optimal schedule $S^\star$ cannot contain the sub-sequence S_2 as well. □

The above result assumes the existence of the parameter k. If such a k does not exist, using sensor i at every time step is optimal. Such a case arises, e.g., when sensor i corresponds to a successful transmission and sensor j corresponds to an unsuccessful one. The issue of optimal sensor scheduling in that case is trivial, unless a bound on the number of times sensor i can be used is given. We shall consider the latter case in the next section.

Thus, we can prune all the branches that include the sequences S_1 and S_2 from the search tree. This gives us a significant decrease in the search space. However, the number of branches still remains exponential in the horizon length K. For a very large value of the horizon K, the complexity is still prohibitive. However, the case for a large enough K is practically identical to considering an infinite horizon cost. For the infinite-horizon cost, we have the following periodicity result that allows us to bypass the tree-search process altogether.

Proposition 6.4.2. *Consider the problem formulation stated in Section 6.2. Suppose that the optimal encoding and decoding functions, as identified in Section 6.4 are being followed. Further, assume that the steady-state has been reached, so that the error covariance in estimating the state $x(m+1)$ based on all the measurements from both the sensors till time m is $P^\star$. Let the sensors be denoted by i and j. Suppose there exists $k > 0$ such*

that

- *For* $m = 1, ..., k-1$, $Trace(h_i^m(P^\star)) \leq Trace(h_j(P^\star))$
- $Trace(h_i^k(P^\star) > Trace(h_j(P^\star))$

Consider the optimal schedule for the infinite horizon case. Suppose that at time step m*, sensor* j *is used. Further, let* $n > 0$ *be the smallest value such that at time* $m + n$*, sensor* j *is used again. Then the optimal schedule after time* m *is given by*

$$t(l) = \begin{cases} j & \textit{if } l = m + kn, \qquad k = 0, 1, 2, \cdots \\ \\ i & \textit{otherwise}. \end{cases}$$

Proof. The proof follows in a straight-forward fashion from the fact that sensor j cannot be used twice in succession due to Proposition 6.4.1[2]. Thus, every time the sensor j is used, the error covariance is 'reset' to $h_j(P^\star)$. Thus, if there is an alternative schedule at time $m + n$ that yields lesser cost, that schedule can be followed at time m to obtain a cost lower than that obtained using the optimal schedule. Thus, the optimal schedule is periodic. □

Using this result, we can solve the optimal scheduling problem for a large horizon in case of a finite-horizon problem, or for the infinite-horizon problem. We solve the finite-horizon problem for a moderate value of the horizon using as the initial covariance $P^\star$. This allows us to obtain the steady-state periodic schedule. Using this result we can obtain the schedule for large values of the horizon. In our experience, moderate values of the horizon $K = 10$ were enough to obtain periodic schedules.

[2]Note that Proposition 6.4.1 was proven for the finite-horizon case. However, since the horizon was arbitrary, the result holds for the infinite-horizon case as well.

6.5 Scheduling a single sensor with a bound on the number of transmissions

The general framework considered in the previous sections facilitates the analysis of a single sensor scheduling in the presence of a bound on the number of transmissions. As argued in the previous section, in the case of a single sensor the issue of scheduling is trivial, unless there is a bound on the number of transmissions. Considering such bounds are important in applications which involve a trade-off between the accuracy of the estimate and the costs of using the sensors and communicating the information to the estimator. In this section we address this issue.

The problem set up is as before except that now we only consider a single sensor observing the process. As before we assume that the steady-state has been reached. For the finite horizon case, denote the length of the horizon by K and the number of allowed transmissions by $c(K) < K$. Therefore the frequency of transmission is defined as:

$$q_K = \frac{c(K)}{K}.$$

We consider the finite horizon problem of selecting the $c(K)$ time instants such that $t(k) = 1$. We denote the choice of 'not to transmit' at time k by $t(k) = \emptyset$. The algorithm for optimal encoding in this case reduces to the sensor maintaining and transmitting an estimate $\hat{x}(k)$ of the state $x(k)$ based on the measurements $y(0), y(1), \cdots, y(k)$. The process estimator updates its estimate $\hat{x}_{dec}(k)$ based on whether it receives new data using

$$\hat{x}_{dec}(k) = \begin{cases} A\hat{x}(k) & \text{if } t(k) = 1, \\ A\hat{x}_{dec}(k-1) & \text{if } t(k) = \emptyset. \end{cases}$$

Consequently, the error covariance at the decoder evolves as:

$$P(k) = \begin{cases} P^{\star} & \text{if } t(k) = 1, \\ AP(k-1)A^T + Q & \text{if } t(k) = \emptyset, \end{cases}$$

where $P^{\star}$ is the steady state error covariance of the optimal estimate of the state $x(k)$ using all the measurements $y(0), y(1), \cdots, y(k-1)$.

We are interested in the following problem: Starting from an arbitrary time m when the last update happened, find which schedule minimizes the cost function

$$\sum_{k=1}^{K} P(m+k) \tag{6.7}$$

subject to the fact that maximum number of the channel use is limited to $n = c(K)$. The following statement indicates that periodic transmission minimizes the cost function.

Proposition 6.5.1. *Consider the problem formulation as stated above. Further, suppose that $j = \frac{K-n}{n+1}$ is an integer. Then, the schedule that minimizes the cost function*

$$\sum_{k=1}^{K} trace(P(m+k)) \tag{6.8}$$

is the periodic schedule

$$t(k) = \begin{cases} 1 & \textit{if } k = m + i(j+1), \quad i = 1, 2, ..., n \\ \emptyset & \textit{Otherwise.} \end{cases}$$

Proof. Consider the sequence $\{P_k\}_{k=1}^{C}$, where

$$P_k = AP_{k-1}A^T + R_W \tag{6.9}$$

with $P_0 = P^\star$ and C being a positive integer greater than 1. Since $P^\star < AP^\star A^T + R_W$, the above-mentioned sequence is increasing in the sense that $P_m < P_n$, where m and n are positive integers such that $m < n$. Denote $T_0 = 0$ and $T_i = \sum_{k=1}^{i} P_i$, $\forall i \in \{1, 2, ..., K\}$.

Note that, every time the sensor transmits, the error covariance at the decoder is reset to $P_0 = P^\star$. Otherwise, it is updated as $P(k) = AP(k-1)A^T + R_W$.

Now consider an arbitrary schedule in which the updates happen at n times $m + t_1, m + t_2, ..., m + t_n$. Define $t_0 = m$ and $t_{n+1} = m + K + 1$. For this schedule the cost function is equal to:

$$\sum_{k=1}^{(n+1)j+n} P(m+k) = nP_0 + \sum_{i=1}^{n+1} T_{t_i - t_{i-1} - 1} = nP_0 + \sum_{i=1}^{n+1} T_{l_i} \tag{6.10}$$

in which $l_i = t_i - t_{i-1} - 1$ is the length of the interval between the ith and $i-1$th transmissions. l_1 is the length of the time interval before (and excluding) the first transmission time and l_{n+1} is the length of the time interval after the last transmission and before K+1. In fact for $i = 1, 2, ..., n$, at the times $m + t_i$ the covariance is reset to P_0. This explains the term nP_0. The terms $T_{t_i - t_{i-1} - 1}$ take care of the cost at the time instances which fall into the "idle" intervals.

So we end up with the following minimization problem:

$$\begin{aligned}
&\min_{l_i} \textstyle\sum_{i=1}^{n+1} T_{l_i} \\
&subject \quad to: \\
&\textstyle\sum_{i=1}^{n+1} l_i = (n+1)j = K - n \\
&T_p = \textstyle\sum_{i=1}^{p} P_i, \quad \forall p \in \{1, 2, ..., (n+1)j + n\} \\
&T_0 = 0 \\
&P_1 < P_2 < ... < P_{(n+1)j+n}
\end{aligned} \tag{6.11}$$

Therefore the problem is to find the optimal assignment of $p_i \in \{0, 1, ..., (n+1)j+n\}$ to l_i in a way that the sum $\sum_{i=1}^{n+1} l_i$ is preserved to be equal to $K-n$. We verify that by keeping the idle interval lengths and therefore the T_{l_i} equal, the cost function is minimized. i.e. $l_i^* = j$, and the minimum cost equals $nP_0 + (n+1)T_j$.

To show this, we first show that if there exist two idle intervals with lengths l_1 and l_4 and $l_1 \neq l_4$, then the cost can be decreased by substituting these two intervals, with two other idle intervals with lengths l_2 and l_3, and shifting the intervals in between so that the length of the other intervals remain unchanged if $l_1 < l_2 < l_3 < l_4$ and $l_2 + l_3 = l_1 + l_4$. The decrease results from the fact that the contribution from the other intervals does not change because their length is preserved. Furthermore,

$$\begin{aligned}
T_{l_1} &= \textstyle\sum_{i=1}^{l_1} P_i \\
T_{l_2} &= T_1 + \textstyle\sum_{i=l_1+1}^{l_2} P_i \\
T_{l_3} &= \textstyle\sum_{i=1}^{l_3} P_i \\
T_{l_4} &= T_3 + \textstyle\sum_{i=l_3+1}^{l_4} P_i
\end{aligned} \tag{6.12}$$

Therefore the only change in the cost incurs as a result of the change in the specific

two intervals. The change in the cost function is equal to:

$$(T_{l_2} + T_{l_3}) - (T_{l_1} + T_{l_4}) = \sum_{l_1+1}^{l_2} P_i - \sum_{l_3+1}^{l_4} P_i < 0 \tag{6.13}$$

This is because the two sums have equal number of elements. Furthermore, because of the monotonicity of the P_i each term in the first sum is less than the corresponding term in the second sum and so the change in the cost is negative. Therefore starting from any two intervals and exchanging the lengths in the above-mentioned manner decreases the cost. The minimum cost corresponds to the case in which no two intervals can be substituted. This is obviously the case when all the intervals are of equal length. So the result follows. □

Remark: If j is not an integer, the time intervals between the sensors cannot be all made equal to j. However, as shown in the proof of the proposition, by choosing the intervals as close to periodic as possible we can get the lowest possible cost.

6.6 Simulation results

In this section we illustrate the results, starting with the improvement in estimation cost using preprocessing. We consider the case of a simple model of two sensors trying to locate a noncooperative vehicle moving in a plane. The model was developed in [47]. The acceleration is equal to zero except for a small perturbation. Let p denote position and v denote speed. Then $x = [p_x \quad p_y \quad v_x \quad v_y]^T$ is the state and we consider a discretization

step h. Following the framework of Section 6.2 the state space model parameters are:

$$A = \begin{pmatrix} 1 & 0 & h & 0 \\ 0 & 1 & 0 & h \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, B = \begin{pmatrix} h^2/2 & 0 \\ 0 & h^2/2 \\ h & 0 \\ 0 & h \end{pmatrix},$$

$$C = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}.$$

The discretization step h is considered to be 0.2 for the simulations. Furthermore, the values of the process and sensor covariances are considered to be

$$R_w = \begin{pmatrix} 0.0100 & 0 \\ 0 & 0.0262 \end{pmatrix},$$

$$R_1 = \begin{pmatrix} 0.0003 & 0 \\ 0 & 0.0273 \end{pmatrix}, \quad R_2 = \begin{pmatrix} 0.0018 & 0 \\ 0 & 0.0110 \end{pmatrix}.$$

Our first observation is that for *all* schedules, preprocessing lowers the cost. The amount of such decrease depends on the particular choice of a sensor schedule. Figure 6.1 shows a histogram of the distribution of this decrease for a small time horizon K=15. It can be seen that more than half of the schedules will incur an improvement of 15% or more.

We also compared the optimal schedules determined with and without preprocessing for different time horizons. The optimal schedule using preprocessing always has a lower cost. Figure 6.2 shows the percentage of the decrease in optimal estimation cost

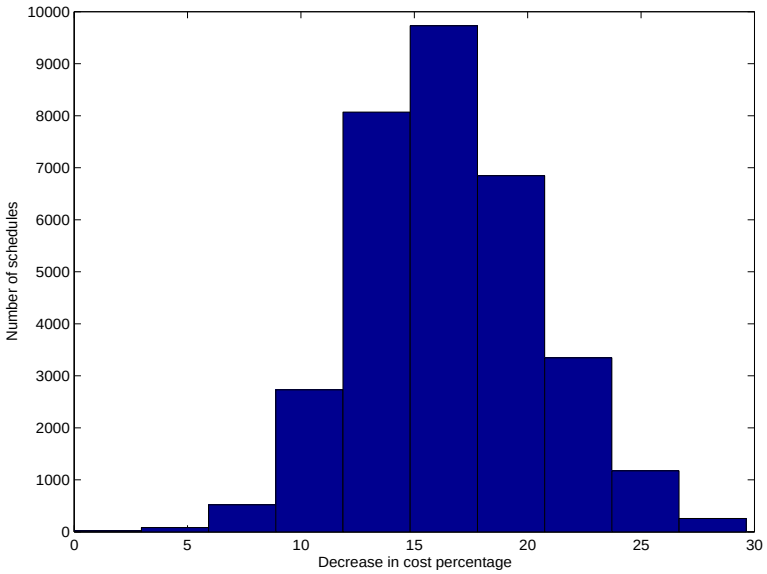


Figure 6.1: Histogram of the percentage of decrease in J_K due to preprocessing. $(K = 15)$

due to preprocessing. We can see that even in this simple system, preprocessing results in more than 18% decrease in estimation cost.

It is worthwhile to note that the optimal schedule has a periodic structure as the horizon increases. The optimal schedule for different horizons are given in table 6.6. The trend remains the same for the values of $k \geq 20$.

The proposed pruning method of section 6.4 results in speed up in the search associated with the scheduling problem. We have measured this by the MATLAB stopwatch timer commands 'tic' and 'toc' for the corresponding tree search routines. This is illustrated in Figure 6.3, where the ratio of the reduction in the CPU time is plotted for the range of horizon $K \leq 15$.

Figure 6.4 illustrates the case of a single sensor S_2. Here a time horizon of $K =$

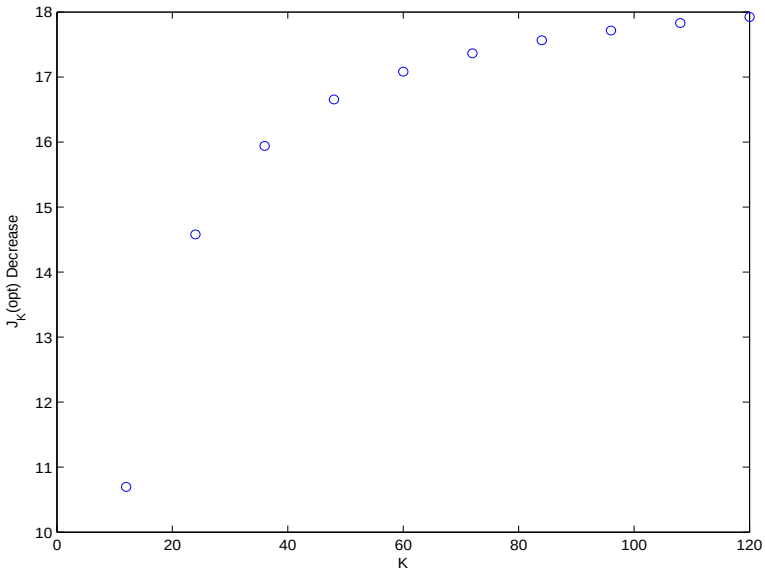


Figure 6.2: Percentage of decrease in J_K for optimal schedule $(k \leq 120)$

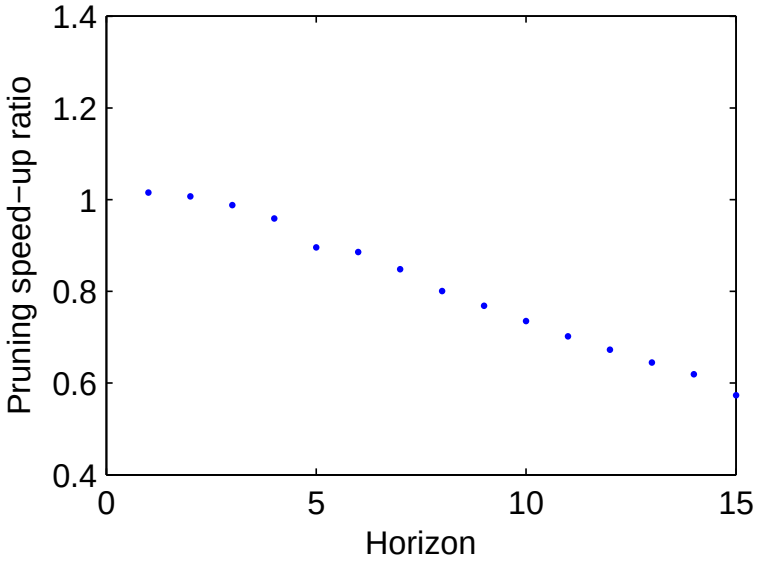


Figure 6.3: CPU time reduction by pruning for $K \leq 15$

K	$OptimalSchedule$
10	2212212212
11	22122122122
12	221221221222
13	2212212212212
14	22122122122122
15	221221221221222
16	2212212212212212
17	22122122122122122
18	221221221221221222
19	2212212212212212212
20	22122122122122122122

Table 6.1: Optimal Schedules

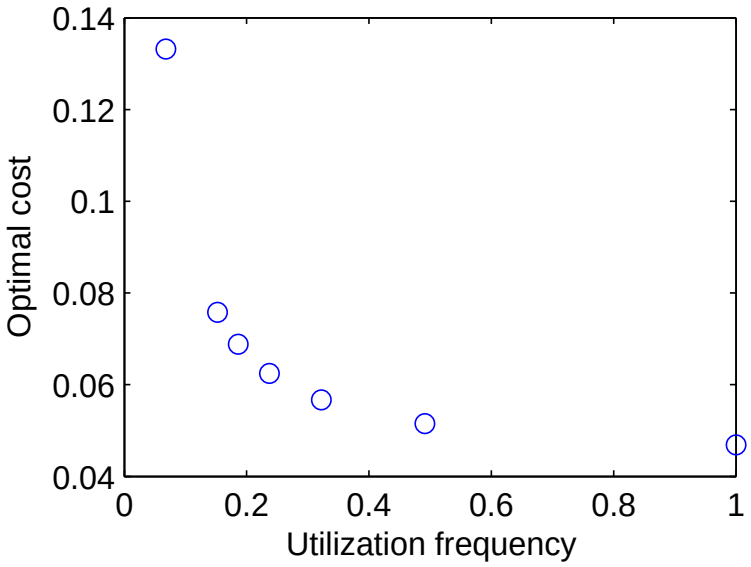


Figure 6.4: Optimal cost in the single sensor case as a function of transmission frequency

59 is considered and the optimal cost is plotted as a function of utilization frequency. $K = 59$ is selected since this particular K results in j being integer for many choices of n. The estimation cost (error) is a decreasing function of sensor utilization. Therefore, in real applications a trade off analysis between the communication and estimation costs determines the frequency of sensor utilization.

Chapter 7

Conclusion

In this dissertation, several aspects of design for networked systems are addressed. The main focus is on combining approaches from system theory with graph theory to characterize graph topologies which result in efficient decision making and control. In this framework, modelling and design of sparse graphs, which are robust to failures and provide high connectivity is considered.

We discuss different factors which affect the performance of a network of autonomous agents. A decentralized approach to path generation in a collaborative system is modelled using potential functions. Inspired from natural swarms, different behaviors in the system such as target following, moving in cohesion and obstacle avoidance is addressed by appropriate encoding of the corresponding costs in the potential function and using gradient descent for minimizing the energy function. Different emergent behaviors emerge as a result of varying the weights attributed with different components of the potential function. The approach uses only local and static information for path generation and therefore, has the benefit of simplicity. It is also flexible and robust, which is important in complex dynamic environments. The disadvantage of the approach is the existence of local minima. Artificial perturbation can help to resolve this problem. An extension of this work uses a Gibbs sampler based randomized algorithm to circumvent this problem [121]. In the continuation of this work, we intend to incorporate practical communication

channels into our model. As a step in this path, in [93] we consider an adaptive distributed policy-based routing scheme for a wireless communication infrastructure that enables a multi-robot system to accomplish its assign tasks. The scheme establishes estimates of link and path costs in the network and constructs probabilistic routing tables in the nodes.

Consensus problems are addressed as a unifying theme in many collaborative control problems and their robustness and convergence properties are studied. We study the implications of the continuous convergence property of the consensus problems on their reachability and robustness. We also investigate the effects of link and agent faults on consensus problems. In particular the concept of invariant nodes has been introduced to model the effect of nodes with different behaviors from regular nodes. A fundamental association is established between the structural properties of a graph and the performance of consensus algorithms running on them. This leads to development of a rigorous evaluation of the topology effects and determination of efficient graph topologies.

It is well known that graphs with large diameter are not efficient as far as the speed of convergence of distributed algorithms is concerned. A challenging problem is to determine a minimum number of long range links (shortcuts), which guarantees a level of enhanced performance. We investigate this problem in a stochastic framework. We specifically study the small world model of Watts and Strogatz and showed that adding a few long range edges to certain graph topologies can significantly increase both the rate of convergence for consensus algorithms and the number of spanning trees in the graph. The simulations are supported by analytical stochastic methods inspired from perturbations of Markov chains. This approach is further extended to a probabilistic framework for understanding and quantifying the small world effect on consensus convergence rates:

Time varying topologies, in which each agent nominally communicates according to a predefined topology, and switching with non-neighboring agents occur with small probability is studied. We provide a probabilistic framework, as well as fundamental bounds on the convergence speed of consensus problems with probabilistic switching. Thereby, we provide a basic procedure for characterizing the small world phenomenon. The results are also extended to the design of robust topologies for distributed algorithms. This work also provides means to better understand the effect of probabilistic switching in consensus and other distributed algorithms. As an immediate next step, the tightness of the bounds presented needs to be characterized for various distributions for switching signals. This will provide the designer with information on how good a time-invariant "average" graph can be as an approximate representor of a time-varying sequence of graph topologies. In a broader context, the relationship to expander graphs and design of methods for generating expander graphs will be studied.

Considering small world networks as graphs for which local connectivity is modelled as a grid and global connectivity is provided by shortcuts, we also address the design of a semi-distributed two-level hierarchical network clustering scheme, which improves the performance of distributed algorithms. The scheme is based on the concept of social degree and local leader selection and the use of consensus-type algorithms for locally determining topology information. Future work includes adjusting our algorithm towards a fully distributed implementation.

Another important aspect of performance in collaborative systems is for the agents to send and receive information in a manner to minimize process costs, such as estimation error and the cost of control. We address an instance of this problem by considering a col-

laborative sensor scheduling problem. In networked control systems, sensors are usually equipped to communicate over wireless channels or communication networks. Thus, it is reasonable to assume that they possess some storage and processing capabilities. Therefore, if the sensors can execute simple recursive algorithms to process the information being collected, significant improvement in estimation (or control) performance can be expected. We use such information processing algorithms along the lines of the ones proposed in [48] for the sensor scheduling problem and show that the optimal algorithms for the sensor scheduling problem require much less data communication than the general multi-sensor problem. Using these information processing algorithms, we show that we obtain significantly better estimates. We consider the problem of finding the optimal sensor schedule. While the general solution remains a tree-search, we show that the number of paths to be searched are significantly pruned. We also prove a periodicity result in the optimal sensor schedules. As a future direction, we will consider the extension of this work to arrays of large sensors and the effects of underlying graph topology on this problem.

A.1 Graphs

Let $G = (V, E)$ be a graph with the set of vertices (nodes) $V = \{v_1, ...v_n\}$, the set of edges $E \subseteq V \times V$. The order of graph is the number of its nodes, n. Let a_{ij} be an indicator of existence of an edge between nodes v_i and v_j, i.e. $a_{ij} = 1$ if an edge exists between V_i and v_j, and $a_{ij} = 0$ otherwise. The n by n matrix $A = [a_{ij}]$ is called the adjacency matrix of graph G. An edge of G is denoted by $e_{ij} = (v_i, v_j)$. If the graph is undirected $e_{ij} \in E$ implies that $e_{ji} \in E$, however for directed graphs there is no such implication. The set of neighbors of node v_i is the set of all nodes for which $a_{ij} = 1$ and is denoted by $N_i = \{v_j \in V | (v_i, v_j) \in E\}$. Degree of a node is the number of its neighbors.

Let $\bar{\mathbb{G}} = \{G_1, G_2, ..., G_m\}$ denote the set of all possible interaction graphs defined for a group of agents. The union of a set of graphs $\{G_{i1}, G_{i2}, ..., G_{im}\}$ each with the vertex set V, is a graph with the vertex set V and the edge set equal to the union of the edge sets of graphs. We say that such a set of graphs is jointly connected if the union of its members is a connected graph. A path between vertices i and j in graph G is a sequence of edges $e_{i,i_1}, e_{i_1,i_2}, ..., e_{i_k,j}$. A graph is said to be strongly connected if there is a path between any of its two vertices. Vertex v_i is said to be linked to vertex v_j across a time interval if there exists a directed path from v_i to v_j in the union of interaction graphs in that time interval. A vertex i of a directed graph is called a root if for any other vertex j there is a path from i to j. A directed tree is a directed graph for which every vertex except the root has exactly one parent. A spanning tree of a directed graph is a tree formed by graph edges that connect all the vertices of the graph. The condition that a graph contains a spanning tree is equal to the condition that all vertices communicate for undirected graphs. For the

directed it is equal to the condition that there exists a vertex having a directed path to all other vertices. We will say the graph G is rooted in i if there i is a root of G. Thus G is rooted in i only if any other vertex of G is reachable from i along a path in the graph. G is called strongly rooted at i if there is an arc from i to every other vertex in the graph. A rooted graph is a graph which has at least one root. A strongly rooted graph is a graph which is strongly rooted in at least one vertex.

The incidence matrix F of an oriented graph with n vertices and l edges is the $n \times l$ matrix such that F_{ij} is equal to 1 if the edge j "enters" vertex i, -1 if edge j exits node i, and 0 otherwise. The symmetric matrix $L = FF^T$ is called graph Laplacian and can be shown that is independent of the orientation chosen for the graph. Furthermore it can be shown that $L = D - A$ and for $x \in R^n$, $x^T Lx = x^T FF^T x = \sum_{(i,j)\in E}(x_i - x_j)^2$. The Laplacian is a semi-positive definite matrix with a zero eigenvalue. The multiplicity of the zero eigenvalue is equal to the number of connected subgraphs of the graph. The first nonzero eigenvalue λ_2, also known as Fiedler eigenvalue is a measure of connectivity of a graph and gives bounds on the vertex and edge connectivity of the graph. Sum of eigenvalues of L is equal to $Tr(L) = Tr(D)$ which is twice the edge number of the graph. If we associate a positive number W_i to each edge and Denote $W = Diag(W_i)$, the matrix $L_W(G) = FWF^T$ is called a weighted Laplacian which satisfies many of Laplacian properties and for $x \in R^n$, $x^T L_W x = x^T FWF^T x = \sum_{(i,j)\in E} a_{ij}(x_i - x_j)^2$, where a_{ij} is the weight corresponding to the edge between vertices i and j. [77] includes a detailed survey of results about Laplacian of graphs and [40] is a good reference of algebraic graph theory.

A.2 Non-negative matrices

A matrix $F \in R^{n\times n}$ is called non-negative if all of its entries are non-negative. In this case we write $F \geq 0$. If a non-negative matrix F has a positive eigenvector, the corresponding eigenvalue is the spectral radius of F, that is, $\rho(F) = \max\{|\lambda_i| : \lambda_i \in \sigma(a)\}$. A non-negative matrix F is called irreducible if and only if $(F + I)^{n-1}$ is positive. For irreducible matrices$\rho(F) > 0$ is a simple eigenvalue corresponding to a positive eigenvector, however, they might also be other eigenvalues with maximum modulus. A non-negative irreducible matrix is called primitive(irreducible and aperiodic) if it has a single eignevalue with maximum modulus. A matrix F is primitive if and only if there exists $N > 0$ such that $F^N > 0$. If all the diagonal elements are strictly positive, then the matrix is primitive if and only if $F^{n-1} > 0$. If $F \geq 0$ is primitive then $\lim_{N\to\infty} \frac{A^N}{\rho(A)^N} = \pi d^T$ where π, d are the right and left eigenvectors associated with the eigenvalue $\rho(F)$ and are both positive.

A non-negative matrix is called stochastic if all row sums are equal to 1. A necessary and sufficient condition for a non-negative matrix to be stochastic is that **1** is its fixed point, i.e. **1** be a right eigenvector corresponding to $\lambda_1 = \rho(F)$. This can be restated as F is a stochastic matrix if and only if $F\mathbf{1} = 1$. A stochastic matrix whose powers converge to a rank-one matrix $\mathbf{1c}^T$ for some vector **c** is called ergodic. All primitive matrices are ergodic but the reverse is not true.

A set of matrices $\Sigma = \{F_1, ...F_m\}$is called Left Convergent Product (LCP) if any left infinite product of the matrices from the set converge to a limit. Σ is said to be para-

contractive with respect to a vector norm $||.||$, if $\forall F_p \in \Sigma, \theta \in R^n$

$$F_p\theta \neq \theta \Rightarrow ||F_p\theta|| < ||\theta|| \tag{1}$$

Reference [101] contain a detailed treatment of Nonnegative matrices.

Bibliography

[1] N. Alon. Eigenvalues and expanders. *Combinatorica*, 6:83–96, 1986.

[2] M. Athans. On the determination of optimal costly measurement strategies. *Automatica*, 18:397–412, 1972.

[3] R. J. Aumann. Agreeing to disagree. *Annals of Statistics*, 4(6):1236–1239, 1976.

[4] D. Babic, D. J. Klein, I. Lukovits, S. Nikolic, and N. Trinajstic. Resistance-distance matrix: a computational algorithm and its application. *International Journal of Quantum Chemistry*, 90:166–176, 2002.

[5] R. Bachmayer and N. E. Leonard. Vehicle networks for gradient descent in a sampled environment. In *Proceedings of the 41st IEEE Conference on Decision and Control*, pages 112–117, Las Vegas, NV, 2002.

[6] J. S. Baras and T. Jiang. Cooperative games, phase transitions on graphs and distributed trust in manet. *Proceedings of 43rd IEEE Conference on Decision and Control*, 1:93–98, 2004.

[7] J. S. Baras, X. Tan, and P. Hovareshti. Decentralized control of autonomous vehicles. *Proceedings of the 42nd IEEE Conference on Decision and Control*, 2:1532–1537.

[8] J.S. Baras and A. Bensoussan. Optimal sensor scheduling in nonlinear filtering of diffusion processes. *SIAM Journal on Control and Optimization*, 27(4):786–813, 1989.

[9] D. Bauso, L. Giarre, and R. Pesenti. Attitude allignment of a team of uavs under decentralized information structure. *Proceedings of IEEE Conference on Control Applications*, pages 486–491, 2003.

[10] D. Bertsekas. *Nonlinear programming*. Athena scientific, 1999.

[11] D. Bertsekas and J. Tsitsiklis. *Parallel and distributed numerical methods*. Athena scientific, 1997.

[12] W.J. Beyn and L. Elsner. Infinite products and paracontracting matrices. *Electronic Journal of Linear Algebra*, 2:1–8, 1997.

[13] V. Blondel, C. de Kerchove, E. Huens, and P. Van Dooren. Social leaders in graphs. *Submitted to POSTA06*, 2006.

[14] V. Blondel, J. Hendrix, A.Olshevsky, and J. Tsitsiklis. Convergence in multiagent coordination, consensus and flocking. *Proceedings 44th IEEE Conference on Decision and Control*, pages 2996–3000, 2005.

[15] B. Bollobas. *Modern Graph Theory*. Springer Verlag, 1998.

[16] V. Borkar and P. Varaiya. Asymptotic agreement in distributed estimation. *IEEE transactions on automatic control*, AC-27(3):650–655, 1982.

[17] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah. Randomized gossip algorithms: Design, analysis and applications. *IEEE Transactions on Information Theory*, 52(6):2508–2530, 2005.

[18] P. Bremaud. *Markov chains, Gibbs Fields, monte Carlo Simulations and Queues*. Springer Verlag, 2nd edition, 1999.

[19] A. L. Bertozzi C. M. Topaz and M. E. Lewis. A nonlocal continuum model for animal aggregation. *Bulletin of mathematical biology*, 68(7):1601–1623, 2006.

[20] M. Cao, A. S. Morse, and B. D. O. Anderson. Reaching a consensus in a dynamically changing environment: A graphical approach. *SIAM Journal on Control and Optimization*, 47(2):575–600, 2008.

[21] R. Carli, F. Fagnani, A. Sperazon, and S. Zampieri. Communication constraints in the average consensus problem. *Automatica*, 44(3).

[22] S. Chatterjee and E. Seneta. Towards consensus: some convergence theorems and repeated averaging. *Journal of Applied Probability*, 14(1):89–97, 1977.

[23] F. Chung. *Spectral Graph Theory*. CBMS Regional Conference Series in Mathematics; American Mathematical Society, 2nd edition, 1994.

[24] F. Chung and L. Lu. *Complex Graphs and Networks*. CBMS Regional Conference Series in Mathematics; Number107 , American Mathematical Society, 2nd edition, 2006.

[25] C. J. Colbourn. *The combinatorics of network reliability*. Oxford University Press, 1987.

[26] F. Cucker and S. Smale. Emergent behavior in flocking. *IEEE Transactions on Automatic Control*, 52(5):852–862, 2007.

[27] G. Cybenko. Dynamic load balancing for distributed memory multiprocessors. *Journal on Parallel and Distributed Computing*, 7:279–301, 1989.

[28] A. K. Das and M. Mesbahi. K-node connected power efficient topologies in wireless networks: asemidifinite programming approach. *Proceedings of the IEEE Global Telecommunications Conference, Globecom*, 1:CD–ROM, 2005.

[29] I. Daubechies and J. Lagarias. Sets of matrices all infinite products of which converge. *Linear Algebra Appl.*, 161:227–263, 1992.

[30] P. J. Davis. *Circulant Matrices*. Wiley, 1979.

[31] M.P. Desai and V.B. Rao. A new eigenvalue bound for reversible markov chains with applications to the temperature-asymptotics of simulated annealing. *Proceedings of the IEEE International Symposium on Circuits and Systems*, 2:1211–1214, 1990.

[32] M. Dorigo and T. Stützle. *Ant Colony Optimization*. "A Bradford book.",The MIT press, 2004.

[33] P. Duchon, N. Hanusse, E. LEbhar, and N. Schabanel. Could any graph be turned into a small world. *Theoretical Computer Science*, 355(1):96–103, 2006.

[34] R. Durrett. *Random Graph Dynamics*. Cambridge University Press, 2007.

[35] F. Fagnani and S. Zampieri. Randomized consensus algorithms over large scale networks. *Proceedings of Information theory and application workshop*, pages 150–159, 2007.

[36] L. Fang and P. Antsaklis. On communication requirements for multi-agents consensus seeking. *Proceedings of Workshop NESC05, Lecture Notes in Control and Information Sciences (LNCIS), Springer*, 331:53–68, 2006.

[37] Lei Fang, Panos Antsaklis, and Anastasis Tzimas. Asynchronous consensus problems: Preliminary results, simulations and open questions. *Proceedings of IEEE Conference on Control Applications*, 2005.

[38] A. Ghosh and S. Boyd. Growing well-connected graphs. *Proceedings of the 45th IEEE Conference on Decision and Control*, pages 6605–6611, 2006.

[39] A. Ghosh, S. Boyd, and A. Saberi. Minimizing effective resistance of a graph. *Proceedings of the 17th International symposium on the Mathematical Theory of Networks and Systems(MTNS)*, pages 1185–1196, 2006.

[40] C. Godsil and G. Royle. *Algebraic graph theory*. Springer Graduate texts in Mathematics, Newyork, 2001.

[41] D. M. Gordon. Control without hierarchy. *Nature*, 446(8):143, 2007.

[42] N. Goyal, L. Rademacher, and S. Vempala. Expanders via random spanning trees. *Proceedings of SODA 2009*, pages 576–585, 2009.

[43] M. H. De Groot. Reaching a consensus. *Journal of American Statistical Association*, 69(345):118–121, 1974.

[44] V. Gupta, T. H. Chung, B. Hassibi, and R. M. Murray. A stochastic sensor selection algorithm with applications in sensor scheduling and sensor coverage. *Automatica*, 42(2):251–260, 2006.

[45] V. Gupta, A. F. Dana, J. Hespanha, R. M. Murray, and B. Hassibi. Data transmission over networks for estimation and control. *IEEE Transactions on Automatic Control*, August 2009. To appear.

[46] V. Gupta, B. Hassibi, and R. M. Murray. Optimal lqg control across packet-dropping links. *System and Control Letters*, 56(6):439–446, 2007.

[47] V. Gupta, C. Langbort, and R. M. Murray. On the robustness of distributed algorithms. *Proceedings of the 45th IEEE Conference on Decision and Control*, pages 3473–3478, 2006.

[48] V. Gupta, N. Martins, and J.S. Baras. Observing a linear process over packet erasure channels using multiple sensors: Necessary and sufficient conditions for mean-square stability. *Proceedings of 46th IEEE Conference on Decision and Control*, 2007.

[49] V. Gupta, N. Martins, and J.S. Baras. Stabilization over erasure channels using multiple sensors. *IEEE Transactions on Automatic Control*, 2007.

[50] D.L. Hall and J. Llinas. An introduction to multisensor data fusion. *Proceedings of the IEEE*, 85(1):6–23, 1997.

[51] Y. Hanato and M. Mesbahi. Agreement over random networks. *Proceedings of IEEE Conference on Decision and Control*, 2004.

[52] F. Harary. A survey of the theory of hypercube graphs. *Comput. Math. Applic.*, 15(4):277–289, 1988.

[53] D. J. Hartfiel. *Nonhomogeneous Matrix products*. World Scientific, 2002.

[54] K. Herring and J. Melsa. Optimum measurements for estimation. *IEEE Transactions on Automatic Control*, 19(3):264–266, 1974.

[55] D.J. Higham. A matrix perturbation view of the small world phenomenon. *Siam Journal on Matrix Analysis and Applications*, 25(2):429–444, 2003.

[56] S. Hoory, N. Linial, and A. Wigderson. Expander graphs and their applications. *Bulletin of the American MAthematical Society*, 43(4):439–561, 2003.

[57] A. Jadbabaie, J. Lin, and A. S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*, 48(6):988–1001, 2003.

[58] A. Jadbabaie, N. Motee, and M. Barahona. On the stability of the kuramoto model of coupled nonlinear oscillators. *Proceedongs of the American Control Conference*, pages 4296–4301, 2004.

[59] Peng Jia and Peter E. Caines. Auctions on networks: Efficiency, passivity, consensus, rate of convergence. *Submitted to 48th IEEE Conference on Decision and Control*, 2009.

[60] M. Juvan and B. Mohar. Laplace eigenvalues and bandwidth-type invariants of graphs. *Journal of Graph Theory*, 17(3):393 – 407, 1993.

[61] S. Kagami and M. Ishikawa. A sensor selection method considering communication delays. *Proceedings of the 2004 IEEE International Conference on Robotics and Automation*, 1:206–211, 2004.

[62] T. Kailath, A. H. Sayed, and B. Hassibi. *Linear Estimation*. Prentice Hall, 2000.

[63] S. Kar and J. M. F. Moura. Topology for global average consensus. *Fortieth Asilomar Conference on Signals, Systems, and Computers (to appear in proceedings)*, 2006.

[64] A. K. Kelmans. Comparison of graphs by their number of spanning trees. *Discrete Mathematics*, 16:241–261, 1976.

[65] A. K. Kelmans. On graphs with maximum number of spanning trees. *Random structures and Algorithms*, 9:177–192, 1996.

[66] A.K. Kelmans. Transformations of a graph increasing its laplacian polynomial and number of spanning trees. *European Journal of Combinatorics*, 18:35–48, 1997.

[67] T.H. Kerr and Y. Oshman. Further comments on 'optimal sensor selection strategy for discrete-time estimators'[and reply]. *IEEE Transactions on Aerospace and Electronic Systems*, 31:1159–1167, 1995.

[68] H. K. Khalil. *Nonlinear systems*. Prentice Hall Inc., 2nd edition, 1996.

[69] V. Krishnamurthy. Algorithms for optimal scheduling and management of hidden markov model sensors. *IEEE Transactions on Signal Processing*, 60(6):1382–1397, 2002.

[70] S. Lee, D. Levin, V. Gopalakrishnan, and B. Bhattacharjee. Backbone construction in selfish wireless networks. *SIGMETRICS*, pages 121–132, 2007.

[71] N. E. Leonard and E. Fiorelli. Virtual leaders, artificial potentials and coordinated control of groups. In *Proceedings of the 40th IEEE Conference on Decision and Control*, pages 2968–2973, Orlando, FL, 2001.

[72] G. Mahoney, W. J. Myrvold, and G. C. Shoja. Generic reliability trust model. *Proceedings of the third annual conference on privacy, security and trust (PST)*, 2005.

[73] I. Matei, N. Martins, and J.S. Baras. Almost sure convergence to consensus in markovian random graphs. *Proceedings of 47th IEEE Conference on Decision and Control*, pages 3535–3540, 2008.

[74] L. Meier, J. Peschon, and R. Dressler. Optimal control of measurement subsystems. *IEEE Transactions on Automatic Control*, 12:528–536, 1967.

[75] C. D. Meyer. Sensitivity of the stationary distribution of a markov chain. *Siam Journal on Matrix Analysis and Applications*, 15(3):715–28, 1994.

[76] B.M. Miller and W. J. Runggaldier. Optimization of observations: a stochastic control approach. *SIAM Journal of Control and Optimization*, 35(5):1030–1052, 1997.

[77] B. Mohar. The laplacian spectrum of graphs. *Graph theory, combinatorics and applications*, 2:871–898, 1991.

[78] L. Moreau. Stability of multi-agent systems with time dependent communication links. *IEEE Transactions on Automatic Control*, 50(2):169–182, 2005.

[79] A. S. Morse. Unpublished manuscript. 2005.

[80] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Phys. Rev. E*, page 02611, 2004.

[81] M. E. J. Newman, C. Moore, and D. J. Watts. Mean-field solution of the small-world network model. *Phys. Rev. Lett.*, 84:3201–3204, 2000.

[82] R. Olfati-Saber. Ultrafast consensus in small-world networks. *Proceedings of American Control Conference*, 4:2371–2378, 2005.

[83] R. Olfati-Saber. Flocking for multi-agent dynamical systems: Algorithms and theory. *IEEE Transactions on Automatic Control*, 51(3):401–4203, 2006.

[84] R. Olfati-Saber. Algebraic connectivity ratio of ramanujan graphs. *Proceedings of the American Control Conference*, pages 4619–4624, 2007.

[85] R. Olfati-Saber and R. M. Murray. Distributed cooperative control of multiple vehicle formations using structural potential functions. In *Proceedings of the 15th IFAC World Congress*, Barcelona, Spain, 2002.

[86] R. Olfati-Saber and R. M. Murray. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on Automatic Control*, 49:1520–1533, 2004.

[87] Y. Oshman. Optimal sensor selection strategy for discrete-time state estimators. *IEEE Transactions on Aeronautics and Electronic Systems*, 30:307–314, 1994.

[88] J. K. Parrish and L. Edelstein-Keshet. Complexity, pattern, and evolutionary trade-offs in animal aggregation. *Science*, 284:99–101, 1999.

[89] H. V. Parunak, M. Purcell, and R. O'Connel. Digital pheromones for autonomous coordination of swarming UAV's. In *Proceedings of AIAA 1st Technical Conference and Workshop on Unmanned Aerospace Vehicles, Systems, and Operations*, 2002.

[90] K. M. Passino. Biomimicry of bacterial foraging for distributed optimization and control. *IEEE Control Systems Magazine*, 22(3):52–67, 2002.

[91] S. Patterson, B. Bamieh, and A. El Ebadi. Distributed average consensus with stochastic communication failures. *Proceedings of 46th IEEE Conference on Decision and Control*, pages 4215–4220, 2007.

[92] D. Peleg. Local majority voting, small coalitions and controlling monopolies in graphs: A review. *Proceedings 3rd Colloquium on Structural information and Communication Complexity*, 1996.

[93] P. Purkayastha, P. Hovareshti, and J. S. Baras. A policy-based routing scheme for multi-robot systems. *Proceedings of the 16th Mediterranean Conference on Control and Automation*, pages 970–975, 2008.

[94] C. Rago, P. Willet, and Y. Bar-Shalom. Censoring sensor: a low communication rate scheme for distributed detection. *IEEE Transactions on Aerospace and Electronic Systems*, 28(2):554–568, 1996.

[95] W. Ren, R. W. Beard, and T. W. McLain. Coordination variables and consensus building in multiple vehicle systems. *Proceedings of the Block Island Workshop on Cooperative Control, Lecture Notes in Control and Information Sciences series,Springer-Verlag*, 309:171–188, 2004.

[96] C. Reynolds. Flocks, herds, and schools: A distributed behavioral model. *Computer graphics*, 21:25–34, 1987.

[97] E. Rimon and D. E. Koditschek. Exact robot navigation using artificial potential functions. *IEEE Transactions on Robotics and Automation*, 8(5):501–518, 1992.

[98] S.I. Roumeliotis and G.A. Bekey. Distributed multi-robot localization. *IEEE Transactions on Robotics and Automation*, 18(5):781–795, 2002.

[99] A. Savkin, R. Evans, and E. Skafidas. The problem of optimal robust sensor scheduling. *Proceedings of the 39th IEEE Conference on Decision and Control*, 4:3791–3796, 2000.

[100] P. Schlegel. The explicit inverse of a tridiagonal matrix. *Mathematics of Computation*, 24(111):665, 1970.

[101] E. Seneta. *Nonnegative Matrices and Markov Chains*. Springer, 2nd edition, 1981.

[102] J. J. E. Slotine, W. Wang, and K. El Rifai. Synchronization in networks od nonlinearly coupled continuous and hybrid oscillators. *Proceedings of the Sixteenth International Symposium on Mathematical Theory of Networks and Systems (MTNS)*, 2004.

[103] D. P. Spanos and R.M. Murray. Robust connectivity of networked vehicles. *Proceedings of the 43rd IEEE Conference on Decision and Control*, 3:2893–2898, 2004.

[104] G. W. Stewart and J. Sun. *Matrix perturbation theory*. Academic Press, 1990.

[105] G.W. Stewart. On markov chains with sluggish transients. *Technical report TR-94-77, Institute for advanced Computer studies, Department of Computer Science, University of Maryland College Park*, 1994.

[106] G. Strang. *Linear Algebra and its Applications*. Academic Press, 2nd edition, 1980.

[107] A. Tahbaz-Salehi and A. Jadbabaie. Small world phenomenon, rapidly changing markov chains, and average consensus algorithm. *Proceedings of 46th IEEE Conference on Decision and Control*, pages 276–281, 2007.

[108] A. Tahbaz-Salehi and A. Jadbabaie. A necessary and sufficient condition for consensus over random networks. *IEEE Transactions on Automatic Control*, 53(3):791–795, 2008.

[109] H. Tanner, A. Jadbabaie, and G. Pappas. Flocking in teams of nonholonomic agents. *Cooperative Control, Springer Lecture Notes in Control and Information Science ,Springer-Verlag*, 309, 2004.

[110] H. G. Tanner, A. Jadbabaie, and G. J. Pappas. Stable flocking of mobile agents, Part I: fixed topology. In *Proceedings of the 42nd IEEE Conference on Decision and Control*, Maui, Hawaii, 2003.

[111] D. Teneketzis and P. Varaiya. Consensus in distributed estimation with inconsistent beliefs. *Systems and Control Letters*, 4:217–221, 1984.

[112] G. Theodorakopoulos and J. S. Baras. On trust models and trust evaluation metrics for ad-hoc networks. *IEEE Journal on Selected Areas in Communications (JSAC)*, 24:318–328, 2006.

[113] F. P. Tsen, T. Sung, M. Lin, L. Hsu, and W. Myrvold. Finding the most vital edges with respect to the number of spanning trees. *IEEE Transactions on Reliability*, 43:600–602, 1994.

[114] J. N. Tsitsiklis. *Problems in decentralized decision making and computation*. PhD thesis, Massachusetts Institute of Technology, November 1984.

[115] J. N. Tsitsiklis and M. Athans. Convergence and asymptotic agreement in distributed decision problems. *IEEE Transactions on Automatic Control*, AC-29(1):42–50, 1984.

[116] S. Vanka, V. Gupta, and M. Haenggi. Power-delay analysis of consensus algorithms on wireless networks with interference. *International Journal of Systems, Control and Communications, Special issue on Information Processing and Decision Making in Distributed Control*, To appear.

[117] T. Vicsek, A. Czirok, E. Ben Jakob, I. Cohen, and O. Schochet. Novel type of phase transitions in a system of self-driven particles. *Phys. Rev. Lett.*, 75:1226–1299, 1995.

[118] D.J. Watts and S.H. Strogatz. Collective dynamics of small-world networks. *Nature*, 393:440–442, 1998.

[119] G. Weichenberg, V. W. S. Chan, and M. Medard. Highly-reliable topological architecture for networks under stress. *IEEE Journal on Selected Areas in Communication*, 22:1830–1845, 2004.

[120] J. Wolfowitz. Products of indecomposable, aperiodic, stochastic matrices. *Proceedings of the American Mathematical Society*, 15:733–736, 1963.

[121] W. Xi, X. Tan, and J. S. Baras. Gibbs sampler-based coordination of autonomous swarms. *Automatica*, 42:1107–1119, 2006.

[122] L. Xiao and S. Boyd. Fast linear iterations for distributed averaging. *Systems and Control Letters*, 53:65–78, 2004.

[123] L. Xiao, S. Boyd, and S. Lall. A scheme for robust distributed sensor fusion based on average consensus. *Proceedings of International Conference on Information Processing in Sensor Networks*, 2005.

[124] M. M. Zavlanos and G. J. Papas. Controlling connectivity of dynamic graphs. *Proceedings of the 44th IEEE Conference on Decision and Control*, pages 6388–6393, 2005.

[125] F. Zhao, J. Shin, and J. Reich. Information-driven dynamic sensor collaboration for tracking applications. *IEEE Signal Processing Magazine*, 19(2):61–72, 2002.