UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE AUTOMAÇÃO E SISTEMAS

Sidney Roberto Dias de Carvalho

NETWORK TOPOLOGY CONTROL FOR CONNECTIVITY MAINTENANCE AND INFORMATION SPREADING MANIPULATION IN MULTI-ROBOT SYSTEMS

Florianópolis (SC)
2020

Sidney Roberto Dias de Carvalho

NETWORK TOPOLOGY CONTROL FOR CONNECTIVITY MAINTENANCE AND INFORMATION SPREADING MANIPULATION IN MULTI-ROBOT SYSTEMS

Tese submetida ao Programa de Pós-Graduação em Engenharia de Automação e Sistemas da Universidade Federal de Santa Catarina para obtenção do título de Doutor em Engenharia de Automação e Sistemas.

Orientador: Prof. Ubirajara Franco Moreno, Dr.

Florianópolis (SC)
2020

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Carvalho, Sidney Roberto Dias de
Network topology control for connectivity maintenance and information spreading manipulation in multi-robot systems / Sidney Roberto Dias de Carvalho ; orientador, Ubirajara Franco Moreno, 2020.
107 p.

Tese (doutorado) - Universidade Federal de Santa Catarina, Centro Tecnológico, Programa de Pós-Graduação em Engenharia de Automação e Sistemas, Florianópolis, 2020.

Inclui referências.

1. Engenharia de Automação e Sistemas. 2. Controle de Topologia. 3. Consenso. 4. Sistemas Multi-Robô. I. Moreno, Ubirajara Franco. II. Universidade Federal de Santa Catarina. Programa de Pós-Graduação em Engenharia de Automação e Sistemas. III. Título.

Sidney Roberto Dias de Carvalho

NETWORK TOPOLOGY CONTROL FOR CONNECTIVITY MAINTENANCE AND INFORMATION SPREADING MANIPULATION IN MULTI-ROBOT SYSTEMS

O presente trabalho em nível de doutorado foi avaliado e aprovado por banca examinadora composta pelos seguintes membros:

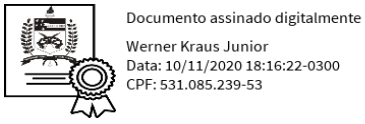
Prof. Ubirajara Franco Moreno, Dr.
PGEAS/UFSC
Presidente

Prof. Jés de Jesus Fiais Cerqueira, Dr.
DEE/UFBA

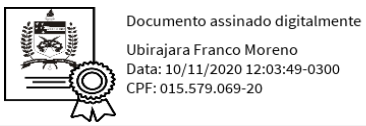
Prof. André Ricardo Fioravanti, Dr.
DMC/UNICAMP

Prof. Eugênio de Bona Castelan Neto, Dr.
PGEAS/UFSC

Certificamos que esta é a **versão original e final** do trabalho de conclusão que foi julgado adequado para obtenção do título de Doutor em Engenharia de Automação e Sistemas.



Prof. Werner Kraus Junior, Dr.
Coordenador do Programa de Pós-Graduação em Engenharia de Automação e Sistemas



Prof. Ubirajara Franco Moreno, Dr.
Orientador

Florianópolis (SC), 09 de Novembro de 2020.

Este trabalho é dedicado à minha família.

Agradecimentos

Agradeço primeiramente a meus pais Roseli e João, por sempre me apoiarem nessa jornada mesmo que à distância. Também aos meus demais familiares: tios, primos, avós que sempre estiveram ao meu lado nos momentos em que mais houve necessidade. Foi através deles, com seus conselhos e orientações, que pude me tornar uma pessoa melhor, persistente em meus objetivos, sem me esquecer de minhas origens.

Agradeço ao meu orientador, o professor Ubirajara Moreno, pela paciência e auxílio durante o desenvolvimento deste projeto, desde o início do mestrado até o final do doutorado. A minha jornada de aprendizado durante esse período foi, em muito, motivada por sua curiosidade e seus questionamentos, sem os quais, certamente, este trabalho não poderia ser concluído.

Agradeço também ao professor Luís Almeida, da Universidade do Porto, pelo seu empenho em me orientar durante o período de doutorado sanduíche em Portugal. Graças ao seu comprometimento com os alunos, pude experimentar um ótimo ambiente de trabalho e um suporte completo durante a minha estadia. Aproveitando esse contexto, gostaria de agradecer a grandes amigos que fizeram toda a diferença durante esse período em terras portuguesas: André Moreira, Inés Alvarez e Luis Pinto. Graças a receptividade e companhia deles, pude viver uma experiência memorável durante tal período.

Agradeço aos amigos de Florianópolis, especialmente àqueles que me acompanharam durante toda a trajetória do mestrado e doutorado, enfrentando dificuldades similares e comemorando juntos as conquistas de cada um, nomeadamente à Alexandre Reeberg, Aujor Tadeu, Fernando Rodrigues, Flávio Gabriel, Guilherme Matiussi, Jonatas Roschild, Karila Palma, Leonardo Salsano, Leonardo Martins, Luis Arcaro, Rejane Barros, Stephanie Brião e Thayse Christine. Assim como também aos "irmãos" de orientação Daniel Ramos, Feres Salem e Renan Tchilian pela convivência diária e pelas discussões aleatórias que tivemos durante esse período.

Gostaria de agradecer também aos amigos de Cuiabá, especialmente à Carla Mendes, Henrique Raia, Lucas Adriano, Luiz Souza, Michel Platini e Ruan Carlos pelas reuniões realizadas rotineiramente e que serviram como uma porta de escape durante os momentos mais difíceis deste processo.

Por fim agradeço a todos aqueles que de alguma forma contribuíram para o desenvolvimento deste trabalho e que não foram explicitamente nomeados, obrigado!

Sidney Roberto Carvalho

"A genuine leader is not a searcher for consensus but a molder of consensus".

– Martin Luther King, Jr.

Resumo

A cooperação é uma propriedade desejada para muitos sistemas interconectados, uma vez que lhes permite resolver tarefas complexas de forma distribuída. O aumento no uso de processos cooperativos (caracterizados pelo compartilhamento de informações) diminui o esforço individual exigido por cada agente. No entanto, esta diminuição do esforço local aumenta a complexidade do sistema em sua totalidade, devido aos acoplamentos intrínsecos entre os agentes que podem originar comportamentos emergentes, os quais são altamente dependentes da topologia de comunicação. Um destes comportamentos, chamado consenso, é necessário para se garantir a coesão entre as informações trocadas. Tal comportamento é essencial para sistemas multi-agentes que são empregados para executar tarefas cooperativas, uma vez que seus indivíduos devem compartilhar informações coesas, que garantam que seus colegas de equipe estejam tentando resolver o mesmo problema. Assim, para manter a cooperação, um sistema interligado deve assegurar a coesão através de dinâmicas de difusão baseadas em consenso. Neste contexto, este trabalho visa dois objetivos distintos ao manipular a topologia de rede de um sistema interconectado composto por agentes autônomos: i) assegurar a conectividade do sistema mesmo quando há falhas em um agente, durante a execução da tarefa cooperativa. ii) controlar a disseminação da informação para dirigir a convergência do consenso, de acordo com a distribuição de informação pretendida para o sistema. Os dois objetivos são abordados separadamente. O primeiro objetivo é resolvido através da solução do Problema do Caixeiro Viajante (TSP) aplicado sobre indicadores de força de sinal em redes sem fio para criar uma topologia virtual bi-conectada em um sistema multiagente. Em seguida, utiliza-se um Controlador Preditivo baseado em Modelo (MPC), executado de forma descentralizada, para aproximar os agentes, transformando os links virtuais em reais e conectando a topologia de rede do sistema. Este procedimento pode transformar qualquer rede conectada em uma rede tolerante a falhas em pelo menos um nodo. Durante o procedimento, uma abordagem baseada em Programação Convexa Sequencial (SCP) aplicada sobre a estrutura do MPC garante a não colisão entre os agentes. Para o segundo objetivo, utiliza-se uma abordagem baseada em Programação Semidefinida (SDP) para se projetar os pesos ótimos da matriz de adjacências da rede, de modo a controlar a convergência de um protocolo distribuído de consenso aleatório para variáveis no domínio de espaço discreto. A teoria de Markov e a inspiração biológica das epidemias são utilizadas para se deduzir um modelo dinâmico de difusão que possa prever como as informações se espalham sob este protocolo de consenso. Além disso, as propriedades de convergência e pontos de equilíbrio do modelo proposto são apresentados em relação à topologia da rede. Finalmente, extensas simulações numéricas e experimentos realizados em uma plataforma robótica comercial avaliam a eficácia das abordagens propostas para ambos os objetivos.

Palavras-chave: consenso, controle de topologia, manutenção da conectividade, difusão de informação, sistemas em rede, design ótimo de pesos.

Resumo Expandido

Introdução

Na robótica móvel, o conceito de trabalho colaborativo ocorre através da necessidade de se realizar tarefas complexas que um único robô não pode resolver. Os sistemas com vários robôs podem executar tarefas usando menos energia por indivíduo e têm maior flexibilidade do que os robôs que trabalham sozinhos. Além disso, a diminuição da complexidade da tarefa resulta em simplificação e barateamento dos robôs usados para resolvê-la. Naturalmente, isso tem um preço: a divisão de tarefas complexas em tarefas mais simples e sua distribuição entre uma equipe de agentes autônomos ou robôs requer cooperação e comunicação. Tais exigências podem ser alcançadas quando os membros do grupo compartilham informações entre eles. Além disso, as informações compartilhadas devem ser coesas em um contexto global. Há, ainda, situações em que a equipe de robôs deve trabalhar sem o apoio de uma estação base ou de uma unidade centralizada, de forma que eles tenham que resolver o problema usando apenas estratégias descentralizadas e informações locais. Todas essas questões são abordadas pela teoria de consenso, que estuda a concepção e análise da dinâmica da informação para forçar o acordo sobre a informação compartilhada em sistemas distribuídos auto-organizados. Outro fator crítico na cooperação dos agentes é a quantidade de comunicação que eles efetuam entre si. Tanto o fluxo quanto a troca de informações são fatores indispensáveis para o sucesso do processo cooperativo. As falhas de comunicação decorrentes de instabilidades causadas pela presença de obstáculos no ambiente ou por falhas nos dispositivos de comunicação podem comprometer a realização das tarefas. Desta forma, a topologia da comunicação e sua manutenção são fatores determinantes para que os robôs possam completar suas missões. A topologia de rede pode ditar toda a dinâmica da informação compartilhada em um sistema interconectado, suprimindo ou espalhando ruídos, dirigindo o comportamento cooperativo do grupo, tornando-o tolerante as falhas, além de outras coisas. Quando um sistema interconectado é um grupo de robôs móveis, outros problemas podem aparecer. Com a mobilidade dos agentes e a limitação de seu alcance de comunicação, o compartilhamento de informações em tais sistemas depende inteiramente da localização espacial dos agentes. Através do ajuste de posição de cada um deles, é possível mudar a topologia da comunicação, melhorando a conectividade do grupo para resolver problemas específicos, ou mudar a forma como eles disseminam a informação no grupo. Na literatura, as abordagens que mudam a topologia da rede e os pesos de suas conexões fazem parte dos estudos sobre controle de topologia.

Objetivos

O principal objetivo de pesquisa desta tese é: controlar a disseminação de informações de um sistema em rede para dirigir sua convergência de consenso, através da manipulação da topologia da rede. Para tanto, são desenvolvidos os seguintes objetivos específicos: (i)

investigar a influência da topologia da rede e do poder social dos agentes na disseminação da informação para um grupo de agentes autônomos sob algum protocolo de consenso em tarefas cooperativas; (ii) propor um modelo que relacione diretamente a dinâmica de consenso do grupo com a difusão da informação e sua topologia de rede subjacente; (iii) manipular a topologia da rede reponderando suas conexões de forma que o consenso sobre uma informação específica seja conduzido para toda a equipe, atingindo um valor desejado pré-definido. Esta tese também possui um objetivo de pesquisa secundário definido como: assegurar a conectividade de um sistema interconectado mesmo quando há falhas em um agente, durante a tarefa cooperativa. Este objetivo secundário é tratado através da extensão de uma abordagem de estado-da-arte apresentada anteriormente para lidar com questões práticas durante a manipulação da topologia de comunicação em um sistema multi-robô, oriundo de dois objetivos específicos: (i) permitir que um sistema multi-robô lide com indicadores de força de sinal das conexões de redes sem fio durante o aumento da conectividade de uma topologia variante no tempo; (ii) evitar colisão entre os robôs móveis durante o deslocamento de posição realizado para aumentar a conectividade de rede de forma descentralizada.

Contribuições

As contribuições deste trabalho são organizadas em dois conjuntos: duas grandes contribuições para as áreas de consenso e manipulação de difusão de informações, e duas contribuições menores para a área de manutenção da conectividade. As contribuições menores são apresentadas no Capítulo 3 e consistem na melhoria de uma abordagem do estado-da-arte para lidar com a variação do nível sinal da conexão, enquanto mantém conectada a topologia de rede do sistema multi-agente durante as tarefas cooperativas. Além disso, uma estratégia ótima para evitar colisões é adicionada à estrutura do algoritmo de controle preditivo baseado em consenso, utilizado nesta abordagem para otimizar o deslocamento dos robôs, garantindo um aumento da conectividade da rede e da área de cobertura ideal destes robôs. As contribuições mais significativas desta tese são apresentadas no Capítulo 4 e consistem em uma abordagem baseada em programação semidefinida e otimização espectral para projetar os pesos ótimos de uma matriz de adjacências de rede, de modo a controlar a convergência de um protocolo de consenso aleatório distribuído para variáveis no domínio do espaço discreto, baseado no Voter Model. Além disso, utilizou-se a teoria de Markov e a inspiração biológica das epidemias para se encontrar um modelo da dinâmica de difusão que possa prever o espalhamento de informações sob este protocolo de consenso discreto e descrever suas propriedades de convergência e pontos de equilíbrio em função da topologia da rede.

Considerações Finais

Este trabalho abordou dois aspectos principais do controle de topologia de rede em sistemas multi-robô: controle de topologia para manutenção de conectividade e controle de topologia para manipulação de difusão de informações. Ambos os aspectos foram explorados independentemente no contexto de agentes autônomos resolvendo tarefas de forma cooperativa, já que este cenário é extremamente dependente das características da topologia de comunicação. As abordagens apresentadas foram avaliadas individualmente em dois cenários distintos: i) um problema de consenso de espaço contínuo para otimização das áreas de sensoriamento em um sistema multi-robô; ii) um problema de consenso de

espaço discreto de agregação distribuída em uma equipe multi-robô. Ambos os cenários compartilham o mesmo núcleo: cooperação distribuída entre indivíduos autônomos. Os resultados deste trabalho aplicam-se principalmente, mas não exclusivamente, aos sistemas multi-robô que seguem alguma estratégia cooperativa para resolver tarefas de forma descentralizada. As tarefas utilizadas em cada cenário devem ser tratadas como um estudo de caso para as estratégias propostas. Pode-se aplicar as ideias apresentadas por esta tese em quase todos os sistemas em rede nos quais os indivíduos devem realizar o compartilhamento de informações e seguir protocolos cooperativos para resolver uma determinada tarefa. Uma destas possíveis aplicações é, por exemplo, as redes sociais cuja dinâmica é dirigida principalmente pela interação entre agentes autônomos (seus usuários, geralmente humanos) através do compartilhamento de informações. Entretanto, as abordagens apresentadas tanto para manter a conectividade em rede quanto para lidar com a disseminação de informações têm limitações e devem, portanto, ser melhoradas de acordo com as exigências de cada tarefa.

Palavras-chave: consenso, controle de topologia, manutenção da conectividade, difusão de informação, sistemas em rede, design ótimo de pesos.

Abstract

Cooperation is a desired property for many interconnected systems since it allows them to solve complex tasks distributively. The increase in the use of cooperative processes (characterised by information sharing) decreases the individual effort demanded by each agent. Nevertheless, this decrease of local effort increases the system complexity as a whole due to the intrinsic couplings between the agents that could give rise to emerging behaviours, which are highly dependent on the communication topology. One of these behaviours, called consensus, is necessary to ensure cohesion among all the exchanged information. This property is an essential and desirable feature in multi-agent systems that are employed to execute cooperative tasks since their individuals must share pieces of information that ensure their teammates are trying to solve the same global problem. Thus, to keep the cooperation, an interconnected system must ensure cohesion through consensus-based diffusion dynamics. In such a context, this work aims two distinct objectives by manipulating the network topology of an interconnected system composed of autonomous agents: i) ensure the system connectivity even when there are faults in one agent, during the performed cooperative task. ii) control the information spreading to drives the consensus convergence, according to the intended information distribution of the system. This thesis undertakes both objectives separately. It solves the first objective through the solution of the Travelling Salesman Problem (TSP) applied over indicators of signal strength in wireless networks to create virtual bi-connected topology in a multi-agent system. Then, it uses a Model Predictive Controller (MPC) executed in a decentralised way to move the agents toward each other, turning the virtual links into real ones and bi-connecting the network topology of the system. This procedure can turn any connected network into a fault-tolerant one. During the procedure, an approach based on Sequential Convex Programming (SCP) applied over the aforementioned MPC framework ensures non-collision among the agents. For the second objective, this thesis uses an approach based on Semidefinite Programming (SDP) to design the optimal weights of a network adjacency matrix, in order to control the convergence of a distributed random consensus protocol for variables at the discrete-space domain. It uses Markov theory and the biological inspiration of epidemics to find out a dynamical spreading model that can predict the information diffusion under this consensus protocol. Also, it presents convergence properties and equilibrium points of the proposed model regarding the network topology. Finally, extensive numerical simulations and experiments performed in a commercial robotic platform evaluate the effectiveness of the proposed approaches for both objectives.

Keywords: consensus, topology control, connectivity maintenance, information spreading, networked systems, optimal weight design.

List of Figures

List of Tables

List of Acronyms

List of Symbols

Notation

– scalars are represented in Roman font, e.g. x, X.
– vectors are represented in lower-case bold face, e.g. $\boldsymbol{x}$.
– matrices are represented in upper-case bold face, e.g. $\boldsymbol{X}$.
– the transpose of a vector $\boldsymbol{x}$ (or matrix $\boldsymbol{X}$) is $\boldsymbol{x}^{\mathsf{T}}$ (or $\boldsymbol{X}^{\mathsf{T}}$).
– the Euclidean norm of a vector $\boldsymbol{x}$ is $\|\boldsymbol{x}\|$ or $\|\boldsymbol{x}\|_2$.
– all vectors are column vectors except when the text says the contrary.
– if $\boldsymbol{x}$ and $\boldsymbol{y}$ are vectors, then $\boldsymbol{x}\boldsymbol{y}^{\mathsf{T}}$ is the matrix product equivalent to the outer product between vectors of the same dimension.
– if $\boldsymbol{x}$ and $\boldsymbol{y}$ are vectors, then $\boldsymbol{x}^{\mathsf{T}}\boldsymbol{y}$ is the matrix product equivalent to the inner product (dot product) between vectors of the same dimension.

Contents

Chapter 1

Introduction

Human challenges in searching for solutions to complex problems show a line of work focused on the distribution of tasks. The division of a complex problem into smaller ones, which can be solved by several cooperating individuals, reduces the initial complexity of such a problem to the issue of cooperation among each teammate aiming to achieve a collective decision that can lead the team to the expected global solution.

In mobile robotics, the concept of collaborative work occurs through the requirement of performing complex tasks that a single robot cannot solve. Multi-robot systems can perform tasks using less energy per individual and has greater flexibility than robots working alone. Furthermore, the decreasing of the task complexity results in simplification and cheapening of robots used to solve it. Naturally, that has a price: the partition of complex tasks into simpler ones and their distribution among a team of autonomous agents or robots requires cooperation and communication. Such requirements can be reached once the teammates share information among them. Also, the shared information must be cohesive in a global context. Besides that, sometimes it is required that the team of robots can work without the support of a base station or a centralised unity, in a way that they have to solve the problem using only decentralised strategies and local information. All those issues are tackled by the consensus theory, that studies the design and analysis of information dynamics to force agreement over distributed information in self-organising networked systems.

Another critical factor in the cooperation of agents is the amount of communication they exchange among themselves. Both the flow and the exchange of information are indispensable factors for the success of the cooperation process. Communication failures arising from instabilities caused by the presence of obstacles in the environment or by faults in the communication devices can compromise the achievement of tasks. In this way, the communication topology and its maintenance are determinant factors for the robots to complete their missions. The network topology can dictate all the dynamics of the shared information in an interconnected system by suppressing or spreading noises, by driving the group's cooperative behaviour, by turning it tolerant to faults, besides other things.

When an interconnected system is a group of mobile robots, other problems might appear. With the mobility of agents and the limitation of their communication range, the information sharing in such systems is entirely dependent on the spatial localisation of the agents. Through the agent's position adjustment, it is possible to change the communication topology, improving the group connectivity to solve specific issues, or to change how they disseminate the information among them. In the literature, the approaches that change the network topology and the weights of its links are part of the topology control studies.

The next sections present some of the most recent and relevant research lines in the field of consensus theory and network topology control on interconnected and cooperative systems. Regarding that, the motivation of this work, its research objectives and contributions are presented in the sequence.

1.1. Related Works

The main research lines related to this work are *the Network Topology Control for Connectivity Maintenance* and the *Network Topology Control for Information Spreading Manipulation*, both applied to cooperative multi-robot systems under consensus protocols. The next sections present the state-of-the-art for those research lines, as well as, for consensus applications in general networked systems. All the presented works in the area of Network Topology Control follow the taxonomy depicted in Figure 1.1. The smaller grey bubbles represent the strategy used to perform one of the considered topology control approaches, and the smaller blue bubbles represent the performed action due to the adoption of its related strategy. Those performed actions result in the desired topology control approach, represented by the larger blue bubbles.

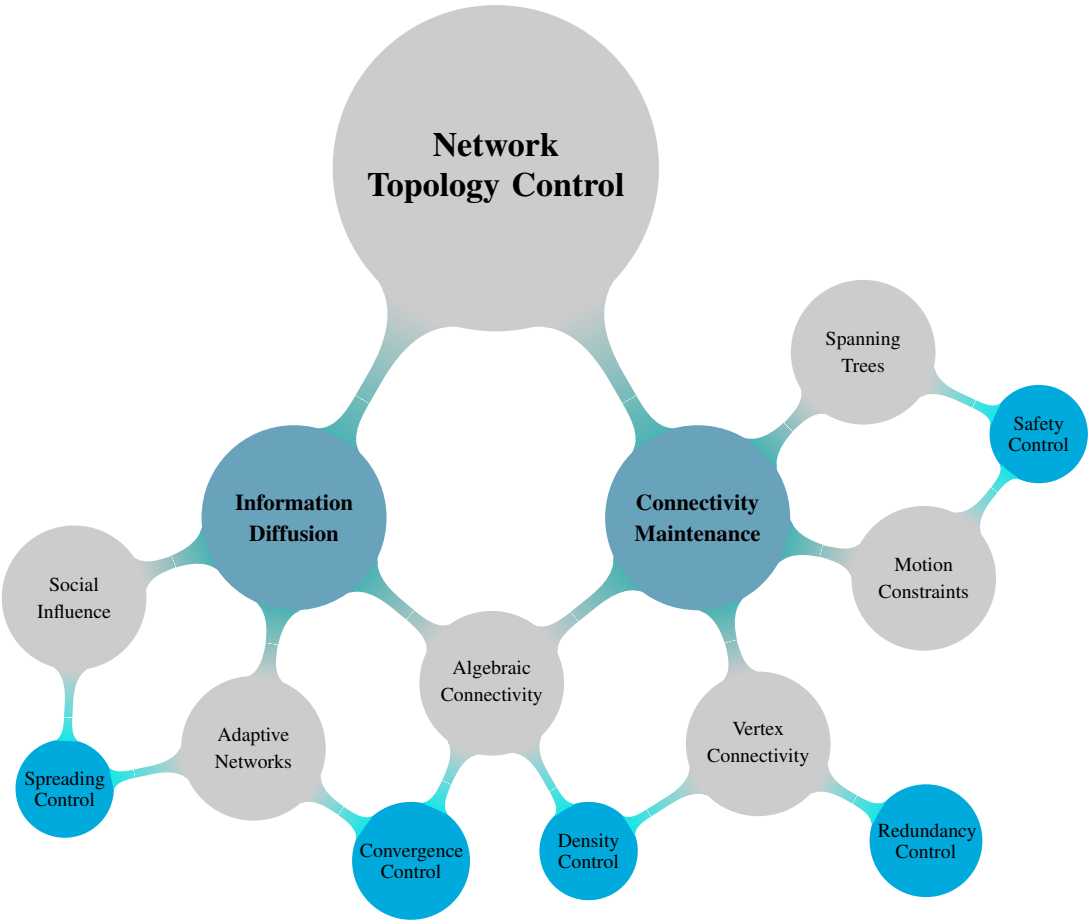


Figure 1.1: Taxonomy of the topology control approaches: the small light grey bubbles describe the theoretical tools and the light blue ones the action performed by the tools connected to them.

1.1.1. Consensus Protocols in Networked Systems

In the last few years, consensus approaches have been extensively studied by the scientific community, mostly because of their capability to deal with information cohesion in decentralised systems and due to the expansion of cheap and intelligent devices that could work autonomously without centralised unit. Such a combination of circumstances turns the implementation of self-organising networked systems to solve complex tasks distributively much more accessible and, consequently, makes the consensus-based approaches very popular.

Consensus approaches are based in the main idea of reach an agreement regarding a certain quantity of interest that depends on the state of all agents by doing local interactions among them. On the literature, the consensus appear primarily, but not exclusively, in averaging algorithms used to compute distributively and cooperatively the global average of each local estimative of a parameter of interest. Those algorithms are present in different contexts, such as synchronisation of coupled oscillators (MIROLLO; STROGATZ, 1990; STROGATZ, 2000; PRECIADO; VERGHESE, 2005), information fusion in sensor networks (KEMPE et al., 2003; OLFATI-SABER; SHAMMA, 2005; XIAO et al., 2005; BOYD et al., 2006), network clock synchronisation (SCHENATO; FIORENTIN, 2011; MAGGS et al., 2012), opinion modelling in social networks (DEGROOT, 1974; MOALLEMI; ROY, 2006; ACEMOGLU et al., 2010; ACEMOGLU; OZDAGLAR, 2011; SALEM et al., 2019), multi-agent coordination and flocking (JADBABAIE et al., 2003; OLFATI-SABER; MURRAY, 2004; OLFATI-SABER et al., 2007; REN; BEARD, 2008), formation control for multi-robot systems (FAX; MURRAY, 2004; CORREIA; MORENO, 2015), to name few. Those approaches are based in the so-called *average consensus protocol*, handled from the perspective of dynamical systems theory.

Besides the average consensus protocol and its derivates, the consensus is present in a bunch of other cooperative approaches used in networked systems. For instance, in recent years the consensus algorithms were extended, by using techniques from control theory, to work with adverse situations such as process and measurement noise (OLFATI-SABER, 2005a; SALEM et al., 2018; TALEBI; WERNER, 2019), exogenous disturbance attenuation (OH et al., 2014; ZHAO et al., 2017; LASHHAB et al., 2017), performance requirements (ORDOÑEZ et al., 2012), time delays (OROSZ et al., 2010), among others. Most of those approaches are based on techniques such as Linear Quadratic Regulators (LQR), Model Predictive Controllers (MPC), Kalman Filters, and Optimal Controllers. The main contribution of such studies, besides the expansion of consensus capabilities, was the decentralisation of those control techniques that traditionally were used in centralised contexts, allowing them to be applicable in multi-agent cooperative systems.

In all those cases, usually, the agreement is performed over variables that belong to a continuous domain. Nonetheless, when distributed cooperation is desired over interconnected systems using variables which represent disjoint and countable information, another kind of approach is required. These variables represent, for example, opinions in social networks, targets in cooperative exploration robots, tasks in processor networks, among others. Such problems are named as *discrete-space consensus* problems, or yet, *consensus decision-making* problems and appear in many fields of science and engineering.

In this context, the first strategies to tackle the discrete-space consensus were based in the so-called quantised consensus, where the states were modelled as discrete values obtained by rounding the continuous variables used in the average-based consensus approaches. That implies, the agreement happens over continuous-space variables, but the decisions that are made regarding this agreement occurs over discrete-space variables.

Some works that used such an approach are (CYBENKO, 1989; KASHYAP et al., 2007) in the context of load balancing in processor networks, (MARTINS, 2008; CHOWDHURY et al., 2016; CERAGIOLI; FRASCA, 2018) in opinion modelling for social networks, and (FANTI et al., 2012; FRANCESCHELLI et al., 2013) in task assignment for distributed systems. Those approaches were promising, but incomplete since they could not handle symbolic values directly due to their mechanism of discretisation of continuous variables.

The first consensus strategies that could work with symbolic variables (e.g. tokens) directly were based in the so-called *Voter Model*, that appeared independently in Physics in the field of probabilistic and interactive particle systems (HOLLEY; LIGGETT, 1975), and in Biology in the field of social competition (CLIFFORD; SUDBURY, 1973). Also, some information diffusion methods that could explore consensus principles over discrete-space were proposed in (GOFFMAN; NEWILL, 1964; KEPHART; WHITE, 1991) using epidemiological inspiration. Those works were seminal. However, they did not fully explore their consensus capabilities in cooperative systems at that time, and only recently, the research community has looked at such results with distributed cooperation in mind.

Concerning that, the consensus over symbolic variables reached great interest from part of the community. For example, the works of Roy et al. (2006), Valentini et al. (2014), Valentini et al. (2017), and Prasetyo et al. (2019) in the context of task assignment and decision making for distributed systems, and the works of Asavathiratham et al. (2001), Yildiz et al. (2010), Banisch et al. (2012), Banisch and Lima (2015), and Noorazar (2020) in opinion dynamics and social modelling, where the handled information are majority symbolic. Those works are prominent in the area, yet have a lot of limitations and drawbacks that still fade the consensus capability of distributed networked systems under discrete-space domain. As, for example, the limit in the amount of unique discrete information that can be analysed simultaneously, and the lack of guarantees due to the nature of the adopted solutions.

1.1.2. Connectivity Maintenance

According to Figure 1.1, the topology control approaches for connectivity maintenance are divided in three sets of applications: *safety control*, that is used to ensure a minimum connected component in the network topology; *redundancy control*, that aims to increase the number of information channels in the network; and *density control*, that aims to increase the number of edges in the network. Each application can be covered through one or more strategies, as shown by the smaller grey bubbles in Figure 1.1.

For safety control in connectivity maintenance for multi-robot systems, the first relevant works appeared around 2001 to 2007. They were mainly based on the minimum connectivity guarantees provided by algorithms built over the minimum spanning tree of the network graph. The works of Li et al. (2003) and Li and Hou (2004), for example, use message exchanges to build a minimum spanning tree of the network and improves the connections of such a tree, ensuring at least a minimum connectivity in a distributed way. On the other hand, the work of Dyer et al. (2007) presents a similar solution, but besides that, it uses the signal strength information provided by the communication interface to improve the vital links of the team, given from the spanning tree of the related graph. Some of the latest works of the literature that rely on building the spanning tree of the communication graph to ensure minimal connectivity are Aranda et al. (2016), to maintain the connectivity of the underlying network under a task of distributed formation control for teams of unicycles robots; Majcherczyk et al. (2018), that presents two strategies to

build a logical spanning tree over the original network in order to keep the connectivity of a very-large robot team; Tardós et al. (2018), that proposes a connectivity maintenance strategy for a coverage area task of heterogeneous robots in a distributed team; and Luo et al. (2019), that presents an algorithm for minimum connectivity maintenance in general cooperative tasks, ensuring the highest freedom for behaviour while it keeps preserved the underlying network connectivity.

The most common approaches used to maintain connectivity through safety control in multi-agent systems found in the literature are based on motion constraints obtained generally, but not exclusively, from specific algebraic graph properties, such as the algebraic connectivity (cf. Chapter 2). For example, the works presented in (ZAVLANOS; PAPPAS, 2005; SCHURESKO; CORTÉS, 2009; YANG et al., 2010; FENG et al., 2015; LI et al., 2020) solve the problem of connectivity maintenance by generating velocities that are assuredly in a set of safety velocities, that cannot disconnect an initially connected network. Those velocities are usually generated by the gradient descent of the Laplacian eigenvalues. Also, some works, such as (GASPARRI et al., 2017; HUNG et al., 2020) use behavioural control to estimate those safety velocities without the need for any spectral information of the network. The main difference among these works is the knowledge scope that is global on centralised approaches such as (ZAVLANOS; PAPPAS, 2005) due to the use of the algebraic properties of the network directly, and local on decentralised or distributed ones such as (SCHURESKO; CORTÉS, 2009; YANG et al., 2010; FENG et al., 2015; LI et al., 2020) that uses the local estimation of the network algebraic properties and (GASPARRI et al., 2017; HUNG et al., 2020) that uses localised behavioural approaches.

Other works follow a different approach, trying to ensure the connectivity maintenance through the maximisation of the algebraic connectivity, obtained from the Laplacian of the underlying network graph and, consequently, increasing the network density. For instance, the works of Kim and Mesbahi (2006), Zavlanos et al. (2011), Fiacchini and Morarescu (2014), and Yang et al. (2019) change the positions of the robots in a way that new links are created among them, and the algebraic connectivity of the graph's Laplacian is maximised, increasing network tolerance to further disconnections. Also, there are works in more general networked systems, such as Sydney et al. (2013) and Khateri et al. (2020), that search for the best edges to be rewired that can increase the algebraic connectivity of the network, improving its robustness.

Another way to ensure connectivity in networked systems through topology control is by increasing the graph's vertex connectivity, which is a network feature intrinsically related to information flow (cf. Chapter 2). This strategy is intimately related to density control, and also to redundancy control. The works of Casteigts et al. (2010), Liu et al. (2010), for example, use virtual forces to build control algorithms that can locally drive each robot, building a bi-connected network that can tolerate at least one node fault without disconnection. Other works, such as (BUTTERFIELD et al., 2008; GHEDINI et al., 2015; CARVALHO et al., 2015; LUO; SYCARA, 2019), use the neighbourhood structure and size to detect each node's criticality, allowing them to improve their connections in order to fix fragile topological configurations and, consequently, turn the network in at least a bi-connected one. Also, there are works such as (SABATTINI et al., 2013; ZAREH et al., 2016b), that employ control strategies to increase the network vertex connectivity through the maximisation of the gradient of the underlying network spectrum. All those approaches usually can be applied without any knowledge about the graph structure or its global algebraic properties, only with local information and position about the neighbours.

1.1.3. Information Spreading Manipulation

In this work, the approaches of topology control for information spreading manipulation are divided in two main applications, according to Figure 1.1: *spreading control*, in which the works for analysis and control of information spreading over networked systems are grouped; *convergence control*, which contains the works that explore the topology control for improvement of the convergence rate of information diffusion algorithms.

The information spreading over networked systems has attracting attention of researchers since early stages of the industrial revolution. At this time, it was especially related to the spreading of epidemics, rumours, and gossips among social individuals. The biological motivations behind those ideas are not by accident. Preliminary studies, such as (GOFFMAN; NEWILL, 1964; KEPHART; WHITE, 1991; PASTOR-SATORRAS; VESPIGNANI, 2001), started proposing to model the information diffusion over interconnected systems using the same mathematical machinery that biologists have been used so far in their models for infectious diseases. In fact, there are a lot of similarities between information dispersion over a networked system and contagious diseases spreading over a population: they are the result of local interaction among autonomous agents and their dynamics is driven by the underlying communication topology.

By using the machinery of mathematical epidemiology, a lot of works proposed approaches to analyse the information diffusion over networked systems, and how the topology influences such spreading over time. The results of Khelil et al. (2002), Eugster et al. (2004), De et al. (2009), Haghighi et al. (2016), and Ojha et al. (2019), for example, relates the average degree of a network to its spreading capability through algebraic graph tools and matrix theory. They apply such concepts through epidemic-based models to have efficient broadcast algorithms that are robust to external attacks. However, in those cases, they do not perform any topology control to improve or change the information spreading.

The topology control to change the information spreading, regarding epidemiological inspiration, is found in works such as (GROSS et al., 2006; SHAW; SCHWARTZ, 2008; GROSS; BLASIUS, 2008; MARCEAU et al., 2010; TRAJANOVSKI et al., 2015; XUE; HIRCHE, 2019a; GUSRIALDI et al., 2019), where they use the so-called co-evolutionary networks (also called adaptive networks) to control information (or disease) spreading over a dynamical network topology through the rewiring of links according to the state of each node. Those approaches allow networked systems to avoid the propagation of specific information (the undesired ones) and boost the propagation of others that are more valuable to the group. Nonetheless, those approaches are hard to analyse since they are time-varying systems in which the network topology is state-dependent and whose dynamics is usually non-linear.

Also, it is recurrent the use of some properties of graphs, as the so-called social influence (also known as social power), to analyse and design more efficient information spreading models in interconnected systems. Works such as (CHEN; GAO, 2012; MIRTABATABAEI; BULLO, 2012; LU et al., 2015; JIA et al., 2017; XIAO et al., 2017) explore the social influence of each node to design topologies that are specialised in the diffusion of certain information. They add more reliability to some nodes (through the increase of their social influence) and less to others (by doing the opposite), suppressing the dissemination of misinformation and boosting the spreading of trustable information. The social power is intimately related to some spectral properties of the underlying network, what implies this adjustment in each node's social influence is made by changing the network topology through rewiring and (or) reweighing of the links.

In consensus applications, the topology control and topology design (i.e. when the topology is not manipulated iteratively over time but built entirely at once) are used widely to improve the consensus convergence rate by increasing the algebraic connectivity of the network, as done, for example, in the works of Xiao and Boyd (2004), Rafiee and Bayen (2010), Dai and Mesbahi (2011), and Kibangou (2012). In those works, they use semidefinite programming and matrix spectral optimisation to design topologies, based in an initially given topology, that increase the convergence rate of the used consensus algorithms (usually average consensus). Also, recent works such as (SHAN et al., 2019; XUE; HIRCHE, 2019b; SHAN et al., 2020), use the rewiring techniques of co-evolutionary networks to mitigate external malicious attacks and external disturbances on consensus-based cooperative systems.

Other works, such as Olfati-Saber (2005b), Kar and Moura (2008), and Liu et al. (2018) use randomised edges' rewiring and reweighing to create time-varying random topologies that increase the consensus convergence speed. It is similar to the strategy of the adaptive network used to control epidemic spreading, but in this case, there is no dependence between the states and the generated network topology. Their capability to increase the consensus convergence rate is due to Ramanujan's and other random graphs properties. Also, such approaches are applied to the spreading control of epidemic (information) on networked dynamical systems, employing the same principles, as found in (RISAU-GUSMAN; ZANETTE, 2009; LAGORIO et al., 2011).

Approaches that use network topology design generally are build in a centralised way, since they are heavily based in spectral optimisation algorithms that must interact with the entire spectrum of the controlled networked systems at once. However, they are more powerful and allow full control of the interconnected system, allowing information spreading manipulation, the improvement on diffusion algorithms' convergence rate, enhancement on the system robustness, among other appealing characteristics. On the other hand, approaches that are based in network topology control (i.e. they change the network topology iteratively over time) are mainly developed as decentralised algorithms and, with that, can be applied more extensively to a broad range of interconnected systems.

1.2. Motivation

As shown in the previous section, the research community interest in the topology control and its impact on the consensus convergence over networked systems has increased a lot in the last years. That is because of the influence of the network underlying topology over consensus and diffusion dynamics. Consensus schemes involve members more deeply and tend to lead to high levels of commitment. Nevertheless, it might be difficult for the group to reach such decisions regarding its size and structure. Also, sometimes it is possible to drive the consensus to a particular value intentionally, by manipulating the network topology, creating the illusion of impartiality and democracy, when in fact there is a bias driving the group's decision.

Those situations are especially problematic when consensus approaches are applied to critical systems in which distributed decision-making protocols are necessary over groups of autonomous agents to solve social decision problems that can compromise the entire stability of the system, such as leader elections, opinion emergence, task distribution, among others. Notwithstanding their vulnerability, consensus approaches still being preferred in these cases because they result in a fairer and equitable decision-making process since they are based on the values of equality, freedom, cooperation and respect

for everyone's needs.

Changing the characteristics of the network topology, such as connections and their weights, in order to manipulate consensus convergence, remains an unsolved problem in the literature, despite the impact of the network topology in the information spreading process required by the consensus protocols. Any changes in the topology have a significant impact over the steady-state of consensus algorithms, but at the same time, they are entirely invisible to each individual in the interconnected system (since they have only local information). That implies those approaches could be used to intentionally drive the consensus behaviour of an entire group of autonomous individuals without being noticed by them.

Also, the topology control for manipulation of consensus would allow one to design network topologies with certain desired characteristics, for example, the improvement of the spreading power of some nodes (who are potentially more trustable) and the decreasing of others (that occasionally spread misinformation). That is applicable not only for multi-robot systems but also on social systems as, for instance, social networks that are currently the most prominent opinion formation tools of human societies. In those systems, misinformation (or disinformation) spreading is a big problem that is too hard to be solved by traditional filtering strategies, and the use of topology control (or topology design) could be more effective to tackle that.

Besides the influence over the information spreading process, the network topology control also must be ensured to be connected to allow the cooperative interaction among the agents, and the topology control for connectivity maintenance, that is extensively explored in the literature, shall be extended to work with more realistic scenarios. For instance, the capability to deal with signal strength variation of wireless networks is desirable on real-world multi-robot systems since those characteristics are more determinant in their communication process than the Euclidean distances often used by the approaches presented in the literature.

1.3. Thesis Objectives

Regarding the motivation of this work and the current state-of-the-art for the consensus theory in self-organising networked systems and network topology control for manipulation of information spreading, the main research objective of this thesis is: ***control the information spreading of a networked system to drives its consensus convergence, through the manipulation of the network topology***. This general objective carries within three specific sub-objectives:

(i) investigate the influence of the network topology and agents social power in the information spreading for a group of autonomous agents under some consensus protocol in cooperative tasks;

(ii) propose a model that directly relates the group's consensus dynamics with the information spreading and its underlying network topology;

(iii) manipulate the network topology by reweighting its links in a way that the agreement over a piece of specific information is driven for the entire team, reaching a pre-defined desired value.

Concerning the topology control for connectivity maintenance, this thesis has a secondary research objective: ***ensure the connectivity of an interconnected system even when there are faults in one agent, during the performed cooperative***

task. This secondary objective is handled by extending a state-of-the-art approach presented earlier to deal with practical issues during the improvement of the communication topology in a multi-robot system, resulting in two sub-objectives:

(i) allow a multi-robot system to deal with signal strength indicators of wireless networks during the connectivity increasing of a time-varying topology;
(ii) avoid collision among mobile robots during the position displacement performed to increase the network connectivity in a decentralised manner.

1.4. Contributions

Concerning this work's objectives and motivation, given the current research scenario, this work's contributions are organised in two sets: two major contributions for the areas of consensus and information spreading manipulation, and two minor contributions for the connectivity maintenance area.

The minor contributions are presented in Chapter 3 and consist of the improvement of the approach early introduced in (CARVALHO et al., 2015; CARVALHO; MORENO, 2015; CARVALHO, 2015) to deal with wireless signal variation while keeping connected the multi-agent network topology during cooperative tasks. Also, an optimal collision avoidance strategy is added to the consensus-based Model Predictive Control framework used previously to optimise the robots' displacement regarding the increase in the network connectivity and the optimal coverage area.

The most significant contributions of this thesis are presented in Chapter 4 and consist of an approach based on semidefinite programming and spectral optimisation to design the optimal weights of a network adjacency matrix, in order to control the convergence of a distributed random consensus protocol for variables at the discrete-space domain, based on the Voter Model. Also, it uses Markov theory and the biological inspiration of epidemics to find out a dynamical spreading model that can predict the information diffusion over this discrete consensus protocol and describe its convergence properties and equilibrium points regarding the network topology.

1.5. Document Organisation

The rest of this document is organised as follows: Chapter 2 presents the theoretical basis of network dynamics and its mathematical representation, including graph theory, matrix theory, and consensus theory. Chapter 3 details the insertion of signal strength sensitivity and collision avoidance into a former topology control approach. Chapter 4 presents and analyse the proposed network reweighting approach to deal with discrete-space consensus spreading's manipulation over networked systems. Finally, Chapter 5 presents the final remarks regarding the presented approaches, as well as possible future research lines that could be explored from this thesis outcomes.

Chapter 2

Network Dynamics

Interconnected systems are everywhere from cell arrangements in living beings, flocks of birds, robotic swarms, to sophisticated social interactions that govern the humankind. All these systems have an essential core in common: communication among individuals. This property allows single agents with simple behaviour to be part of intricate groups and solve tasks cooperatively, that otherwise were impossible to be addressed. Networks are the synthesised representation of interconnected systems and such communication among their fundamental elements. Each networked system has its dynamics by which the information evolves through its components. Such a dynamics governs how the agents cooperate and exchange information with each other allowing, among other things, cohesion between their actions and decisions. These cohesive diffusion dynamics, which can drive the local information of each element to a global value, results in an emergent behaviour called *consensus*.

This chapter introduces the essential concepts of the network structures, algebraic representation of networks, information diffusion dynamics, and consensus theory, as well as some network applications that use consensus-based dynamics to solve decentralised and cooperative tasks.

2.1. Network Structures

Networks are the synthesised version of interconnected systems. They are a symbolic representation of such systems that allow us to solve complex problems and make analysis with high reliability and precision when compared to the real systems. This section presents some fundamental mathematical tools used to describe networks as well as some of their more valuable properties.

2.1.1. Graphs

Graphs are a natural mathematical abstraction to the structures that control the information diffusion in a network. They are mainly used to describe the communication topology of a network through symbolic elements such as edges and vertices.

A finite graph is build over a set of finite elements that represent the nodes of the synthesised network. That set is called in the literature as *vertex set* and normally it is represented by the letter $\mathcal{V}$. The connections among the nodes are called *edges* or *links* and they are part of the *edges set*, commonly represented by $\mathcal{E}$. Formally, a graph is defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where the vertex set is defined as $\mathcal{V} = \{1, \ldots, n\}$ with n integer numbers representing each node, and the edges set is defined as $\mathcal{E} = \{(1,2), \ldots, (i,j)\} \subseteq \mathcal{V} \times \mathcal{V}$, with a pair of nodes (i,j) describing a connection from node i to node j, where $i, j = 1, \ldots, n$.

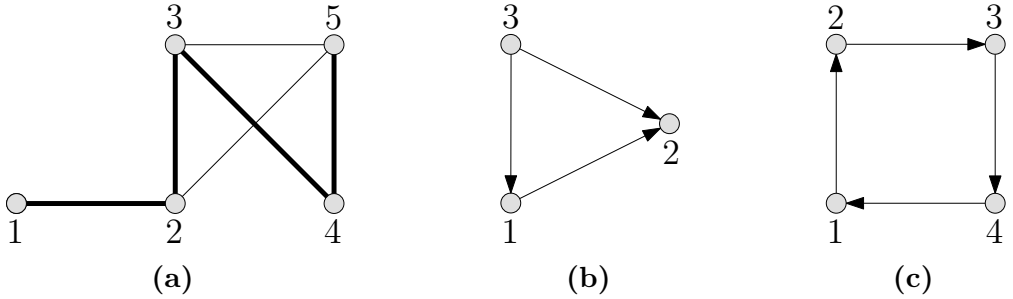


Figure 2.1: Graph examples: (a) undirected graph, (b) and (c) directed graphs. Lines are edges, circles are vertices, and arrows represent the communication direction.

Figure 2.1a is the graphical representation of a undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $\mathcal{V} = \{1, \ldots, 5\}$ and $\mathcal{E} = \{(1,2), (2,1), (2,3), (3,2), (3,4), (3,5), (4,3), (4,5), (5,2), (5,3), (5,4)\}$. Throughout this document, the terms agent, robot, individual, vertex, and node are used interchangeably, similarly, to the terms network and graph, and edge and link.

When there is a relationship between two vertices i and j (i.e. there is an edge between them), they are called adjacent vertices (or neighbours in some contexts[1]), and such relationship is represented as $i \sim j$. One can define the *neighbours set* of a node i as

$$\mathcal{N}_i = \{j \in \mathcal{V} : (i,j) \in \mathcal{E} \text{ or } (j,i) \in \mathcal{E}\} \subseteq \mathcal{V}. \tag{2.1}$$

The size of the neighbours set of a node i is called vertex degree (or valency) and it is represented by $|\mathcal{N}_i|$. If a graph has all its nodes with the same degree, i.e. $|\mathcal{N}_i| = |\mathcal{N}_j|,\ \forall i, j \in \mathcal{V}$, then it is called *regular graph*.

A subgraph $\mathcal{G}' = (\mathcal{V}', \mathcal{E}')$ of a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is defined in function of the subsets of vertices and edges, such that $\mathcal{V}' \subseteq \mathcal{V}$ and $\mathcal{E}' \subseteq \mathcal{E}$. If $\mathcal{V}' = \mathcal{V}$, then $\mathcal{G}'$ is called *spanning subgraph* of $\mathcal{G}$. When such subgraph has no cycles[2], then it is called *spanning tree* (GROSS; TUCKER, 2009). In Figure 2.1a, the bold lines are the edges of a spanning tree.

A path $\mathcal{P}_{ij}$ is a subgraph of a graph $\mathcal{G}$ that have a chain which starts at vertex i and finishes at vertex j. Formally, it is defined as $\mathcal{P}_{ij} = (\mathcal{V}', \mathcal{E}') \subseteq \mathcal{G}$ where $\mathcal{V}' = \{i, \ldots, j\}$ and $\mathcal{E}' = \{(i, i_{+1}), (i_{+1}, i_{+2}), \ldots, (i_{+m/2}, j_{-m/2}), \ldots, (j_{-1}, j)\}$, where i_{+1} means the next vertex after i, j_{-1} is the vertex immediately before j, and $m \leq n$ is the size of $\mathcal{V}'$. A node j is said to be *reachable* if there is path $\mathcal{P}_{ij}$ starting in a node i and finishing in j, in this case one say that such node is reachable from i. A spanning tree is a path that allows at least one vertex to reach all others (GROSS; TUCKER, 2009).

Graphs are classified according to the information flow constraints they have. If there are no restrictions in the information flow, namely the communication is bidirectional, then the graph is called an *undirected graph* (or just graph). On the other hand, if there are directed links, i.e. the information flows from one node to another but not vice-versa, the graph is called a *directed graph* (or digraph).

In digraphs, the set of edges is not symmetrical and such edges represent the link direction. For instance, if $(i,j) \in \mathcal{E}$ (i.e. there is a directed edge from i to j), then i is called the link's *tail* and j is called the link's *head*. The set of tail nodes related to a vertex i (i.e. the set of nodes whose links start on them and finish on i) is called *set of input neighbours* and it is defined as $\mathcal{N}_i^+ = \{j \in \mathcal{V} : (j,i) \in \mathcal{E}\}$, representing the nodes from whom the vertex i receive information. The set of head nodes related to a vertex

[1] In this text, both terms refer to connected nodes.

[2] A cycle in a graph is a non-empty trail in which the only repeated vertices are the first and last one.

i (i.e. the set of nodes whose links start on i and finish on them) is called *set of output neighbours* and it is defined as $\mathcal{N}_i^- = \{j \in \mathcal{V} : (i,j) \in \mathcal{E}\}$, representing the nodes who receive information from i. The general neighbours set can be defined in function of both sets as $\mathcal{N}_i = \mathcal{N}_i^+ \cup \mathcal{N}_i^-$.

In Figure 2.1b a digraph defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $\mathcal{V} = \{1,2,3\}$ and $\mathcal{E} = \{(1,2),(3,1),(3,2)\}$ is depicted. Note that, the set of output neighbours of node 2, $\mathcal{N}_2^- = \{\emptyset\}$. The same to the set of input neighbours of node 3, $\mathcal{N}_3^+ = \{\emptyset\}$. These nodes are called *sink vertex* and *source vertex*, respectively.

When the size of the set of input neighbours is equal to the size of the set of output neighbours for every vertex of the digraph, i.e. $|\mathcal{N}_i^+| = |\mathcal{N}_i^-|,\ \forall i \in \mathcal{V}$, then it is called *balanced digraph*. All undirected graphs are, by definition, balanced. Figure 2.1c, depicts a balanced digraph where $|\mathcal{N}_i^+| = |\mathcal{N}_i^-| = 1,\ i = 1,\ldots,4$. As shown by the next sections, this property of digraphs is of high importance to information diffusion over interconnected systems.

2.1.2. Connectivity

Connectivity is an important property of graphs that allows one to performs robustness analysis, as well as detect congestion points and other interesting characteristics of networks. For instance, by analysing the connectivity of a graph, it is possible to say how many elements (edges and vertices) of such graph can be removed without cause its disconnection.

One way of evaluate the connectivity of a graph is by evaluating the presence or absence of some vertices or edges, the so-called *cut vertices* and *cut edges*, respectively. A cut vertex is a vertex of a connected graph that will disconnect it when removed. Similarly, a cut edge is an edge of a connected graph that can disconnect it if removed (FIEDLER, 1973; KIRKLAND et al., 2015).

A *vertex cut-set* of a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a subset of $\mathcal{V}$ whose removal turns $\mathcal{G}$ into two or more subgraphs that are disconnected from each other. The cardinality of the smaller vertex cut-set of a graph $\mathcal{G}$ is called *vertex-connectivity*, and it is represented by $\kappa_v(\mathcal{G})$. Such graph is called κ_v-vertex-connected or simply κ_v-connected graph. The *edges cut-set* of a graph is a subset of $\mathcal{E}$, and its definition is analogous to the vertex cut-set. The cardinality of the smaller edges cut-set is called *edge-connectivity*, and it is represented by $\kappa_e(\mathcal{G})$. The related graph is called κ_e-edge-connected (GODSIL; ROYLE, 2001). The graph depicted in Figure 2.1a, for instance, has one cut vertex (vertex 2), and two cut edges (the edge $(1,2)$ and $(2,1)$). So, it is called 1-vertex-connected graph and 2-edge-connected graph.

Another way to evaluate the graph's connectivity is through the estimation of the number of spanning trees on it. The number of spanning trees in a graph is a robustness metric of it because the number of distinct paths among all nodes is proportional to the number of spanning trees in the graph. Thus, it is possible to ensure edge-connectivity through the number of spanning trees that a graph has. A digraph is said to be *strongly connected* if it has at least an undirected spanning tree, oppositely, it is called weakly connected. All connected undirected graphs are strongly connected by definition (GROSS; TUCKER, 2009).

From robustness viewpoint, the minimum size of the smallest vertex cut-set (or edge cut-set) necessary to ensure connectivity even when there is a deletion of one vertex (or edge) is 2. When a graph is 2-connected, it is called bi-connected; when it is 3-connected,

it is called tri-connected, and so on. The next chapter presents an approach that turns any arbitrary connected graph into a bi-connected one.

2.1.3. Algebraic Graphs

Graphs can be represented in an algebraic form using matrices. Thus, the concepts of vertex and edge are suppressed and replaced by straight algebraic notations that can be manipulated with linear algebra tools.

The most direct algebraic version of a graph is its *adjacency matrix*. As the name suggests, this matrix describes the adjacency relationships among vertices. The general adjacency matrix of a graph $\mathcal{G}$ is defined as a square integer matrix with n rows and n columns indexed by the n vertices of the graph

$$\boldsymbol{\mathcal{A}}(\mathcal{G}) = [a_{ij}] \in \mathbb{R}^{n \times n},$$

where $a_{ij} = 1$, if $j \sim i$ (or $(j,i) \in \mathcal{E}$), and $a_{ij} = 0$, otherwise. There is also the weighted version of such a matrix, where the values represent the strength of each graph's edge. Note that, the definition of the adjacency matrix used here is in accord to the works in the multi-agent area, such as (OLFATI-SABER; MURRAY, 2003; OLFATI-SABER et al., 2007; MURRAY, 2007; REN; BEARD, 2008; MESBAHI; EGERSTEDT, 2010), being slightly different from the definition used in the field of theoretical graph theory and pure mathematics.

If $\mathcal{G}$ is a strongly connected digraph, then its corresponding adjacency matrix $\boldsymbol{\mathcal{A}}(\mathcal{G})$ is called *irreducible matrix*. Otherwise, it is called a *reducible matrix*. If $\boldsymbol{\mathcal{A}}(\mathcal{G})$ is irreducible, there is an integer $0 < k < \infty$ for every element of $\boldsymbol{\mathcal{A}}^k(\mathcal{G})$ to be greater than zero. In fact, each element at row i and column j of $\boldsymbol{\mathcal{A}}^k(\mathcal{G})$ represents the number of paths starting at node j and finishing at node i with length k (in hops). In other words, $\boldsymbol{\mathcal{A}}(\mathcal{G})$ is irreducible if every node can be reached from any other node in a finite number of hops. On the other hand, if there is an integer $0 < k < \infty$ such that $\boldsymbol{\mathcal{A}}^k(\mathcal{G})$ is strictly positive (i.e. all its elements are simultaneously greater than zero), then $\boldsymbol{\mathcal{A}}(\mathcal{G})$ is called *primitive* (or ergodic) matrix and its related underlying graph $\mathcal{G}$ is a *primitive graph* (GODSIL; ROYLE, 2001).

Another algebraic representation of a graph is its *valency matrix* or *degree matrix*. It is a diagonal matrix containing the degrees of all vertex of the graph

$$\boldsymbol{\Delta}(\mathcal{G}) = \operatorname{diag}(\boldsymbol{\mathcal{A}}\mathbf{1}_n) = \begin{bmatrix} |\mathcal{N}_1| & 0 & \dots & 0 \\ 0 & |\mathcal{N}_2| & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & |\mathcal{N}_n| \end{bmatrix}$$

where $\operatorname{diag}(\cdot)$ creates a diagonal matrix using the input parameter, $\mathbf{1}_n$ is an array of ones with order n, and $|\mathcal{N}_i| \in \mathbb{N}$ is the vertex i's degree. If $\mathcal{G}$ is a digraph, then its degree is the size of the set of input neighbours $|\mathcal{N}_i^+|$, i.e. its in-degree (MESBAHI; EGERSTEDT, 2010).

The graph depicted in Figure 2.1a has the following degree and adjacency matrices

$$\boldsymbol{\Delta}(\mathcal{G}) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 3 \end{bmatrix} \quad \text{and} \quad \boldsymbol{\mathcal{A}}(\mathcal{G}) = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix}.$$

The most valuable algebraic representation of a graph is the *Laplacian matrix*, that is obtained by applying the continuous Laplacian operator over a finite graph (or digraph) $\mathcal{G}$ (MERRIS, 1994; GODSIL; ROYLE, 2001). The Laplacian matrix of a general graph is defined as

$$\boldsymbol{\mathcal{L}}(\mathcal{G}) = [\ell_{ij}] \in \mathbb{R}^{n \times n},$$

where $\ell_{ij} = -a_{ij}$ if $i \neq j$, and $\ell_{ii} = \sum_{j=1}^{n} a_{ij}$. An equivalent representation uses the relationship between the degree matrix and the adjacency matrix, resulting in

$$\boldsymbol{\mathcal{L}}(\mathcal{G}) = \boldsymbol{\Delta}(\mathcal{G}) - \boldsymbol{\mathcal{A}}(\mathcal{G}).$$

For example, the Laplacian matrix of the graph represented in Figure 2.1a is

$$\boldsymbol{\mathcal{L}}(\mathcal{G}) = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 \\ -1 & 3 & -1 & 0 & -1 \\ 0 & -1 & 3 & -1 & -1 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & -1 & -1 & -1 & 3 \end{bmatrix}.$$

Note that, the sum of all elements in a row, for any row, is zero. That implies zero as one of its eigenvalues and $\mathbf{1}_n$ as the related right-eigenvector. As shown next, this property can describe a lot about a Laplacian matrix's underlying graph.

When the text does not directly refer to the graph $\mathcal{G}$ or its explicit definition is not necessary, for simplicity, the formulation drops out the graph's indication from the definition of the corresponding algebraic graph without any loss of representativeness, e.g. $\boldsymbol{\mathcal{A}}(\mathcal{G})$ becomes $\boldsymbol{\mathcal{A}}$.

2.1.4. Spectral Analysis

The algebraic representation of graphs allows one to verify multiple properties of connected systems by using their spectra. This technique is known as *spectral analysis* of networks (NEWMAN, 2010b).

One important spectral property of a graph is given by the *spectral radius* of its adjacency matrix and can be used as a bound to characterise its structure concerning the algebraic eigenvalues. It is defined in function of the supreme among the absolute eigenvalues of the adjacency matrix

$$\rho(\boldsymbol{\mathcal{A}}) = \max\{|\lambda_1(\boldsymbol{\mathcal{A}})|, \ldots, |\lambda_n(\boldsymbol{\mathcal{A}})|\},$$

where $\lambda_i(\boldsymbol{\mathcal{A}})$ is the i-th lower eigenvalue of $\boldsymbol{\mathcal{A}}$. Also, the largest eigenvalue of the adjacency matrix is related to the degree of a general graph $\mathcal{G}$ in the following way

$$\frac{1}{n}\mathbf{1}_n^{\mathsf{T}}\boldsymbol{\mathcal{A}}(\mathcal{G})\mathbf{1}_n \leq \lambda_n(\boldsymbol{\mathcal{A}}) \leq \overline{\delta}$$

where $\overline{\delta} = \max_{i \in \mathcal{V}}\{|\mathcal{N}_i|\}$ is the largest degree of $\mathcal{G}$. If $\mathcal{G}$ is regular, then $\lambda_n(\boldsymbol{\mathcal{A}}) = \overline{\delta}$ (proofs follow Godsil and Royle (2001)).

The eigenvalues of a Laplacian matrix, on the other hand, have some properties that can be used to classify the connectivity degree of the related graph. Such eigenvalues are ordered as

$$0 = \lambda_1(\boldsymbol{\mathcal{L}}) \leq \lambda_2(\boldsymbol{\mathcal{L}}) \leq \ldots \leq \lambda_n(\boldsymbol{\mathcal{L}}).$$

The number of zeros as eigenvalues of $\boldsymbol{\mathcal{L}}(\mathcal{G})$ (i.e. their algebraic multiplicity) is precisely the number of connected components at $\mathcal{G}$ (FIEDLER, 1973; MOHAR, 1991). That implies, for a graph to be connected a necessary and sufficient condition is $\lambda_2(\boldsymbol{\mathcal{L}}) > 0$, i.e. the second smallest eigenvalue of $\boldsymbol{\mathcal{L}}(\mathcal{G})$ must be greater than zero otherwise $\mathcal{G}$ is disconnected. This property is called *algebraic connectivity* (FIEDLER, 1973; GODSIL; ROYLE, 2001).

In recent works, such as (ZAREH et al., 2016a), they found that the third smallest eigenvalue of the Laplacian ($\lambda_3(\boldsymbol{\mathcal{L}})$) can be directly related to the vertex-connectivity. That gives a lower-bound to the connectivity estimative which turns easier the fault tolerant design of networks. Besides that, there are works, such as (BARAS; HOVARESHTI, 2009) relating the $n-1$ larger eigenvalues of the Laplacian with the number of spanning trees in the graph.

An important observation obtained from the characterisation of the Laplacian eigenvalues is

$$\lambda_2(\boldsymbol{\mathcal{L}}) \leq \kappa_v(\mathcal{G}) \leq \kappa_e(\mathcal{G}) \leq \underline{\delta}(\mathcal{G}),$$

where $\underline{\delta}(\mathcal{G}) = \min_{i\in\mathcal{V}}\{|\mathcal{N}_i|\}$ is the minimum degree of the graph $\mathcal{G}$.

2.1.5. Centrality Score

In graph theory and network analysis, the centrality score is a way to measure the node importance in a graph according to its spacial position on there. It is used to evaluate the individual vertex influence on the network dynamics in many applications such as social interactions, consensus protocols, epidemic spreading, and others (NEWMAN, 2010a).

On literature, there are many centrality measures such as degree centrality, closeness centrality, betweenness centrality, harmonic centrality, Katz centrality, eigenvector centrality, to say some. According to the application requirements, a measure can be better than others since a centrality which is optimal for one application is often sub-optimal for a different one. This work is mainly interested in the eigenvector centrality because of its relation with some network algebraic properties, as described in the next sections.

In the eigenvector centrality, nodes whose neighbours have a high out-degree (i.e. a large number of links starting from it) are more influential than others. It implies the node's score is directly determined by the importance of its neighbours, especially the ones who receive information from it (BONACICH, 1972).

The eigenvector centrality is defined directly in terms of the adjacency matrix, as

$$v_i = \frac{1}{\lambda}\sum_{j=1}^{n} a_{ji}v_j \qquad i = 1, \ldots, n$$

that in matrix form results in

$$\lambda\boldsymbol{v} = \boldsymbol{\mathcal{A}}^{\mathsf{T}}(\mathcal{G})\boldsymbol{v} \tag{2.2}$$

where $\boldsymbol{v} = [v_1,\ v_2,\ \ldots,\ v_n]^{\mathsf{T}}$ is the array of individual centrality scores, and $\lambda \in \mathbb{R}$ is a constant.

Note that (2.2) is the eigenvector equation, more precisely the left-eigenvector equation with λ being an eigenvalue of $\boldsymbol{\mathcal{A}}(\mathcal{G})$, and $\boldsymbol{v}$ being its corresponding left-eigenvector. Due to the additional requirement that all the entries in the eigenvector must be non-negative, only the largest eigenvalue $\lambda_n(\boldsymbol{\mathcal{A}})$ and its respective left-eigenvector results in the desired centrality measure. Also, it is common to use the row-normalised adjacency matrix,

$\boldsymbol{\Delta}^{-1}(\mathcal{G})\boldsymbol{\mathcal{A}}(\mathcal{G})$, instead of the regular adjacency matrix to obtain absolute centrality scores (NEWMAN, 2010a).

If $\boldsymbol{\mathcal{A}}(\mathcal{G})$ is unweighted, the eigenvector score can be defined individually for each node i as

$$v_i = \frac{\delta_i}{|\mathcal{E}|} \qquad i = 1, \ldots, n$$

where $|\mathcal{E}|$ is the cardinality of the edges set (i.e. the number of edges in the graph), and $\delta_i = \sum_{j=1}^{n} a_{ij} \triangleq |\mathcal{N}_i^+|$ is the in-degree of node i. Observe that this notation evidences how the neighbourhood size influences in the centrality score of each node.

2.2. Diffusion on Networks

As pointed out in the last section, the main property of networks is their connectivity, which facilitates the circulation of conditions such as states, information, goods, diseases or anything that can be "transmitted" from one node to another. The circulation of some quantity over time is known as diffusion, which is a dynamic process. The diffusion through networks is responsible for the epidemic spread of viruses or diseases, "viral" news, rumours (good and bad), memes and ideas. Therefore, the dynamic properties of dissemination over networks is an important and current topic of interest. Besides that, it shares much in common with cooperative dynamics, which often occurs within networks of autonomous individuals, such as multi-agent and multi-robot systems.

This section examines how network topology – the pattern of who is connected to whom – affects diffusion. Diffusion is one of the first issues for which the network structure is definitively essential. The connection pattern can make the difference between a viral video and an obscure one, or a pandemic infection and a limited outbreak, for example.

2.2.1. Diffusion of Continuous-Space Information

For continuous-space information, the diffusion models describe how continuous values of concentrations or other continuous properties are transported from one node to another through different network structures. These models are often used to build cohesive diffusion dynamics who are responsible for emergent behaviours that leads to distributed agreement, for example.

One of the simplest diffusion models for continuous-space information is based on the so-called *gossip algorithm* and it is build around random selection of peers which average their information. In this algorithm, at every time step k, a node $i \in \mathcal{V}$ is chosen randomly with probability $1/n$ to interact with another node $j \in \mathcal{V}$ also randomly chosen but with probability p_{ij}. Then, both average their information, while the others keep their current information unchanged. Assuming a continuous-space information state $x_i[k] \in \mathbb{R}$ for the selected node i at time step k, the gossip spreading rule is defined as

$$x_i[k+1] = x_j[k+1] = \frac{x_i[k] + x_j[k]}{2} \quad \text{w.p.} \quad p_{ij} \qquad j = 1, \ldots, n \tag{2.3}$$

all other nodes different from i and j keep their current information state, i.e. $x_q[k+1] = x_q[k],\ \forall q \neq i, j$; and $j \in \mathcal{V}$ is a randomly chosen node with probability $p_{ij} \in [0, 1]$. Due to the arbitrary choice of only one node i to interacts with another, also randomly selected, at each time step k, this protocol is named as randomized asynchronous gossip (BOYD et al., 2006).

Usually the probabilities p_{ij} are defined regarding the network topology, as for example in (SHAH, 2007), where those values assume the inverse of the in-degree of each node i, yielding

$$p_{ij} = \frac{a_{ij}}{\delta_i}, \qquad i = 1, \ldots, n$$

i.e. all neighbours of a selected node i are equiprobable to interact with him at a given time step k. Also, sometimes a weighted version of the inverse of the in-degree is used, allowing one to manipulate the information diffusion over the network.

Regarding the definition of p_{ij}, Equation (2.3) might lead or not to a cohesive dynamics that results in agreement (or consensus) over the distributed average of all the initial information states. Next section explores more deepening this kind of cohesive dynamics from which consensus emerges in interconnected systems.

2.2.2. Diffusion of Discrete-Space Information

Models of network diffusion for discrete-space information consider the situation where a local event spreads, or fails to spread, through the network. They are mainly constituted of a spreading rule describing how agents in the network influence one another. The most straightforward rules mirror models of disease spread whereas the diffusion of discrete-space information on networks often evolve from local interaction among their nodes and happens in a very similar way as virus and contagious diseases spread on biological networks (i.e. it has epidemic dynamics) (EUGSTER et al., 2004). In this case, the discrete-space domain for information means that a node can have it or not at some time instant.

Early works, such as (GOFFMAN; NEWILL, 1964; KEPHART; WHITE, 1991), started proposing to model the diffusion of discrete-space information using the same mathematical machinery that biologists have been used so far in their models for infectious diseases. Indeed, many models of information diffusion in networks take inspiration from well-known "compartmental models" of disease spread such as the SI, SIS, SIR, and SEIR models, to name a few. Their names are formed by the initial letters of the terms *Susceptible*, *Infected*, *Exposed* and *Recovered*, which represent abstract compartments where each individual of a population is placed depending on his (or her) state of health (BAILEY, 1975; DALEY; GANI, 1999).

Despite their recurrent use for analysis of information spreading, compartmental models do not correctly capture the spread dynamics of information over heterogeneous networks since they consider the underlying network topology well mixed or full connected, which in both cases presumes the strong assumption of regularity and do not cover the majority of the interconnected systems. To deal with that, some works such as (WANG et al., 2003; GANESH et al., 2005; SHARKEY, 2008; MIEGHEM et al., 2009; MIEGHEM, 2011), presented the so-called *intertwined epidemic models*, a kind of network epidemic model which captures the individual interaction among all the nodes of a network and can describe the spreading of discrete-space information over any interconnected system precisely.

In the intertwined SIS model, for example, each node has two internal states that change according to the possession of particular information: the susceptible state which is assumed by a node when its information state is empty; and the infected state which is assumed by a node when it has the particular information. Those states compose a discrete Markov chain, whose transition probabilities are defined regarding two values: the *infection rate* which describes the rate at which a node becomes infected, and the *recovery*

rate, that represents the rate at which a node forgets about the particular information. It is the simplest model for diseases in which infection does not confer immunity. Thus, susceptible become infected and, then, become susceptible again upon recovery. Note that, this model handles only one information (or virus) in the system per time, so if a node does not have such information, its information state should be empty.

This and other epidemic models are not deterministic at all. They use probabilities to describe the uncertainty over the spread mechanism employed by the nodes to share their information. Through the analysis of the dynamics of such probabilities, one can estimate topological conditions necessary to ensure that particular information may reach every node in the network, becoming epidemic. In this case, the diffusion dynamics are said to be cohesive, and that results in an emergent behaviour called consensus, as stated in Section 2.3. Chapter 4 presents an epidemic modelling of the spreading dynamics of a random discrete-space consensus protocol and uses such a model to allow the manipulation of the information spreading in this protocol through the topology reweighting.

2.3. Agreement and Consensus Theory

One of the fundamental problems in networks is the information cohesion, where all their nodes agree about the value of some parameter. By solving this problem, one can allow complex tasks to be performed by numerous individuals cooperatively, usually faster than when it is solved in a non-cooperative way. The agreement or consensus theory studies how information diffusion dynamics must be designed to ensure, from local estimative, the cohesion of information at a global level (the consensus) for an interconnected system at steady-state. This section introduces the basic concepts of consensus theory, as well as, the mathematical tools used to analyse the convergence of consensus-based algorithms regarding network topology for both continuous-space and discrete-space variables.

2.3.1. Agreement over Continuous-Space Information

Algorithms for consensus over continuous-space information appears when there is a necessity of distributed agreement over variables that belong to a continuous-space domain, in the sense that they cannot be enumerated. It is equivalent to say, for example, that the consensus sample space for continuous-space information is the entire set of real numbers. Thus, the cohesive diffusion dynamics that lead to an agreement over continuous-space variables must operate over the set of real numbers.

An often explored protocol for consensus in continuous-space is the so-called *linear average consensus protocol* that is built mathematically through the weighted average of the information states that describe the agents' local estimative of some parameter of interest (OLFATI-SABER; MURRAY, 2003). The cohesive diffusion dynamics that handle this protocol is built for an node (agent) $i \in \mathcal{V}$ over a continuous-space variable $x_i[k] \in \mathbb{R}$, representing its information at time step k, through the following difference equation

$$x_i[k+1] = x_i[k] + h \sum_{j=1}^{n} (x_j[k] - x_i[k]) a_{ij} \qquad i = 1, \ldots, n, \tag{2.4}$$

where $h \in \mathbb{R}$ is a very small constant related to the integration step-size.

The consensus over the information states is reached when the difference between them is null, i.e. $|x_i[k] - x_j[k]| = 0,\ \forall i, j \in \mathcal{V}$ for some $k > 0$. The state at which it happens is called consensus *equilibrium point* and it is defined as $\lim_{k\to\infty} x_i[k] = x^\star,\ \forall i \in \mathcal{V}$.

Equation (2.4) ensures such a value can be reached asymptotically regarding the network topology, as shown next section.

2.3.1.1. Convergence Analysis

To analyse the behaviour of the linear average consensus protocol, Equation (2.4) must be extended to its matrix form, yielding

$$\boldsymbol{x}[k+1] = \underbrace{(\boldsymbol{I}_n - h\boldsymbol{\mathcal{L}})}_{\boldsymbol{\Phi}}\boldsymbol{x}[k] \tag{2.5}$$

where $\boldsymbol{x} = [x_1,\ x_2,\ \ldots,\ x_n]^\mathsf{T}$ is the array of information states for all nodes in the network, $\boldsymbol{\mathcal{L}}$ is the Laplacian matrix of the underlying network graph, and $\boldsymbol{\Phi} = \boldsymbol{I}_n - h\boldsymbol{\mathcal{L}}$ is a row-stochastic matrix known as *Perron matrix* (OLFATI-SABER et al., 2007).

Equation (2.5) represents the entire network dynamics, and it is useful to perform analysis of convergence and other analysis over the distributed system from a global viewpoint. A usual starting point for such analysis is through the closed solution of the consensus dynamics given in terms of the network initial state as

$$\boldsymbol{x}[k] = \boldsymbol{\Phi}^k\boldsymbol{x}[0] \tag{2.6}$$

which can be written using the spectral decomposition of $\boldsymbol{\Phi}$, resulting in

$$\begin{aligned}\boldsymbol{x}[k] &= \boldsymbol{Q}\boldsymbol{\Lambda}^k\boldsymbol{Q}^{-1}\boldsymbol{x}[0] \\ &= \sum_{i=1}^{n}\boldsymbol{q}_i\lambda_i^k(\boldsymbol{\Phi})\bar{\boldsymbol{q}}_i\boldsymbol{x}[0]\end{aligned} \tag{2.7}$$

where $\boldsymbol{Q} = [\boldsymbol{q}_1,\ \boldsymbol{q}_2,\ \ldots,\ \boldsymbol{q}_n]$ is a matrix whose columns are eigenvectors of $\boldsymbol{\Phi}$, $\boldsymbol{\Lambda} = \mathrm{diag}([\lambda_1(\boldsymbol{\Phi}),\ \lambda_2(\boldsymbol{\Phi}),\ \ldots,\ \lambda_n(\boldsymbol{\Phi})])$ is a diagonal matrix formed by the ascending ordered eigenvalues of $\boldsymbol{\Phi}$, and $\bar{\boldsymbol{q}}_i$ is the i-th row of matrix $\boldsymbol{Q}^{-1}$.

If the constant h is defined in the range $0 < h < 2/\lambda_n(\boldsymbol{\mathcal{L}})$ and $\boldsymbol{\mathcal{L}}$ has at least a directed spanning tree in its structure, then matrix $\boldsymbol{\Phi}$ has all its eigenvalues inside the unit circle and Equation (2.7) is stable and convergent (XIAO; BOYD, 2004; OLFATI-SABER et al., 2007). Otherwise, matrix $\boldsymbol{\Phi}$ has two (or more) maximum eigenvalues equal to 1 (as $\boldsymbol{\mathcal{L}}$ has two zero eigenvalues) and there is no convergence to the global consensus at all. That happens because the steady-state of the information states is defined by the linear combination of the eigenvectors related to the largest eigenvalue of $\boldsymbol{\Phi}$, yielding

$$\begin{aligned}\lim_{k\to\infty}\boldsymbol{x}[k] &= \left(\cancelto{0}{\lambda_1^k(\boldsymbol{\Phi})}\boldsymbol{q}_1\bar{\boldsymbol{q}}_1 + \cancelto{0}{\lambda_2^k(\boldsymbol{\Phi})}\boldsymbol{q}_2\bar{\boldsymbol{q}}_2 + \ldots + \cancelto{1}{\lambda_n^k(\boldsymbol{\Phi})}\boldsymbol{q}_n\bar{\boldsymbol{q}}_n\right)\boldsymbol{x}[0] \\ &= \boldsymbol{q}_n\bar{\boldsymbol{q}}_n\boldsymbol{x}[0]\end{aligned}$$

where $\lambda_n(\boldsymbol{\Phi}) = 1$ is the largest eigenvalue of $\boldsymbol{\Phi}$, $\boldsymbol{q}_n = \mathbf{1}_n/\sqrt{n}$ is the ℓ_2-normalised unit eigenvector related to $\lambda_n(\boldsymbol{\Phi})$, and $\bar{\boldsymbol{q}}_n$ is the left-eigenvector of $\boldsymbol{\Phi}$ related to $\lambda_n(\boldsymbol{\Phi})$. If matrix $\boldsymbol{\Phi}$ is also column-stochastic, what is possible only if matrix $\boldsymbol{\mathcal{L}}$ is symmetric or its underlying digraph is balanced, then $\bar{\boldsymbol{q}}_n$ is also a ℓ_2-normalised unit vector, and the steady-state of $\boldsymbol{x}$ becomes precisely the average of the initial information states

$$\lim_{k\to\infty}\boldsymbol{x}[k] = \left(\frac{1}{\sqrt{n}}\right)^2\mathbf{1}_n\mathbf{1}_n^\mathsf{T}\boldsymbol{x}[0] = \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\mathsf{T}\boldsymbol{x}[0] = \frac{1}{n}\mathbf{1}_n\sum_{i=1}^{n}x_i[0].$$

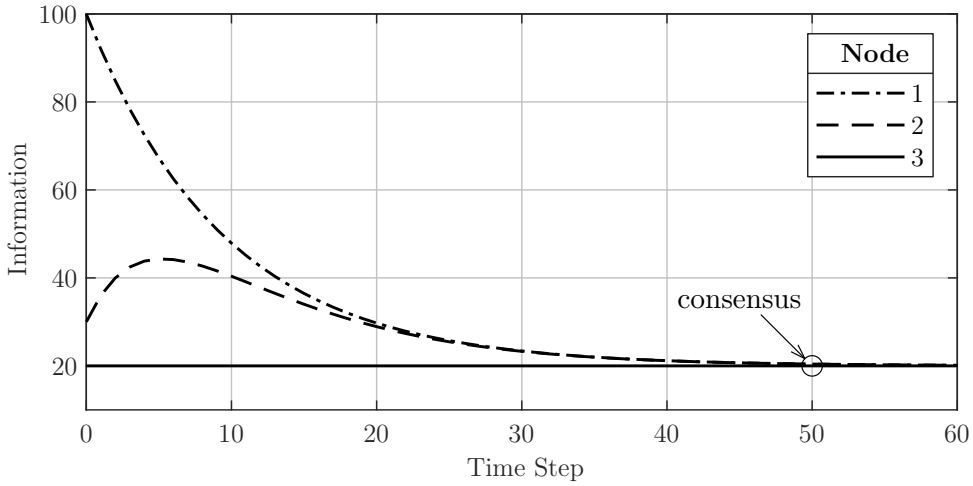


Figure 2.2: Information state trajectory of the average consensus protocol over time for each node.

If $\boldsymbol{\mathcal{L}}$ is not symmetric and its underlying digraph is not balanced, but it still strongly connected, then the consensus equilibrium point becomes a weighted average of the initial information states. In cases where $\mathcal{G}$ is not strongly connected, but it has yet a directed spanning tree in its structure (i.e. one of its nodes is a source vertex), then the consensus is driven by the information state of the root of this tree (the source node), that is the consensus equilibrium point is the information of such a node. All other topological configurations lead to a non convergence to the consensus (OLFATI-SABER et al., 2007).

Example 2.3.1. As example of how the linear average consensus protocol evolves in a multi-agent network, considers the digraph in Figure 2.1b. The initial information states of all nodes is defined as $\boldsymbol{x}[0] = [100\ 30\ 20]^{\mathsf{T}}$. The graph Laplacian matrix and its eigenvalues are

$$\boldsymbol{\mathcal{L}} = \begin{bmatrix} 1 & 0 & -1 \\ -1 & 2 & -1 \\ 0 & 0 & 0 \end{bmatrix} \qquad \lambda(\boldsymbol{\mathcal{L}}) = \begin{bmatrix} 0 \\ 1 \\ 2 \end{bmatrix}.$$

Note that the second smaller eigenvalue of the Laplacian is greater than zero ($\lambda_2(\boldsymbol{\mathcal{L}}) = 1$), so the system meets the algebraic graph requirements to the consensus convergence.

Figure 2.2 depicts the trajectories of the individual information states updated using Equation (2.4). After some simulation time, the nodes reach a common value for their information. According to the dynamics in Equation (2.4), if the digraph was balanced, the equilibrium point should be the approximated average of the initial information states, that is $\frac{100+30+20}{3} = 75$. However, as can be seen, the steady-state is reached when all information is equal to 20, which is, not coincidentally, the initial information of node 3. It happens because, as stated before, when there is a source vertex in the network it drives the consensus value, due to the absence of information from others agents in its dynamics.

2.3.2. Agreement over Discrete-Space Information

When individuals of a group must agree distributively about a particular subject or issue that belongs to a countable or discrete domain, they have to solve the so-called *consensus decision-making problem*, a collaborative decision-making process where a team

of interconnected individuals reasoning about an issue agree to support a decision that leads the whole group to a common goal. Thus, consensus decision-making problems are concerned with the process of deliberating and finalizing decisions cooperatively through information exchange.

The main difference between consensus decision-making and other decision-making strategies is on the unanimity required by the consensus approaches. In engineering and other related fields, these consensus decision-making problems appear in situations where there is a group of autonomous individuals (or agents) that must decide a value for some parameter of interest over a finite set of available options (i.e. a countable set) collectively. When the available set of options is not enumerable, this problem resumes itself to the continuous-space consensus problem.

One of the simplest diffusion dynamics that leads to consensus over discrete-space information is a stochastic process named *Voter Model*, in which for every time step k a node i in the network, selected randomly with probability $1/n$, chooses to adopt the same state of one of its neighbours, j, taken uniformly at random from i's neighbourhood, i.e. with probability $1/\delta_i$, while all other nodes keep their current state unchanged (YILDIZ et al., 2010). Node i is said to be active at time step k, all other nodes are inactive at the same time step. Assuming a discrete-space variable $x_i[k] \in \mathbb{S}$ for the selected node i at time step k, the asynchronous Voter Model can be defined according to the following update rule

$$x_i[k+1] = x_j[k] \quad \text{w.p.} \quad p_{ij} = \frac{a_{ij}}{\delta_i} \qquad j = 1, \ldots, n \tag{2.8}$$

all other nodes different from i keep their current information state unchanged, i.e. $x_q[k+1] = x_q[k],\ \forall q \neq i$; and $p_{ij} \in [0, 1]$ is the probability of node i chooses node j to get its information. The finite set $\mathbb{S}$ is called sample space and contains countable elements which represent the discrete-space information[3] available to the interconnected system. Note that, this equation is very similar to the gossip protocol described in Equation (2.3).

The voter model has been extensively studied in the field of opinion dynamics to model democratic voting processes and spatial conflict between different species (CLIFFORD; SUDBURY, 1973; HOLLEY; LIGGETT, 1975). From an engineering point of view, it has few requirements, since it only demands the selected node at each time step to process the opinion of a neighbour. The voting model leads to particularly precise collective decisions, always reaching consensus if the topology matches with certain requirements, but has long decision times that are proportional to the size and connectivity of the network, as shown next section.

2.3.2.1. Convergence Criteria

As the update sequence is random, and evolution depends only on the most recent update, the whole system can be seen as a Markov chain with N^n states, where $N \in \mathbb{N}$ is the cardinality of the set $\mathbb{S}$, i.e. the number of unique information in the system. Each state of such a Markov chain is then given by a n-dimensional vector, built around the node states. Clearly, the condition in which all nodes have the same information is an absorption state, since a node cannot choose an information that does not appear in the network.

The work of (YILDIZ et al., 2010) proofs that the probability of reaching consensus

[3] In the original Voter Model proposed by (CLIFFORD; SUDBURY, 1973), $\mathbb{S} = \{0, 1\}$, i.e. it could represent only two possible values.

after n interactions between nodes is always greater than zero, i.e.

$$\Pr\{x_i[k+n] = x_j[k+n],\ \forall i,j \in \mathcal{V}\} > 0$$

if the network topology is strongly connected and n is finite. Also, they prove that the Voter Model leads to consensus almost surely since its spreading dynamics is a Markov chain, and its absorbing state is a condition where all nodes have the same information. A sufficient condition for convergence is that there always exists a strictly positive probability of reaching the absorbing state, what is true if the network topology is strongly connected.

The expected convergence time of the generalised Voter Model is upper bounded by the network topology in the work of (YILDIZ et al., 2010), as follows

$$\tau \leq \frac{4e\log(n+2)|\mathcal{E}|}{1-\lambda_{n-1}(\boldsymbol{\Delta}^{-1/2}\boldsymbol{\mathcal{A}}\boldsymbol{\Delta}^{-1/2})}\max_{i\in\mathcal{V}}\delta_i^{-1}$$

where $\lambda_{n-1}(\boldsymbol{\Delta}^{-1/2}\boldsymbol{\mathcal{A}}\boldsymbol{\Delta}^{-1/2})$ is the second largest eigenvalue of matrix $\boldsymbol{\Delta}^{-1/2}\boldsymbol{\mathcal{A}}\boldsymbol{\Delta}^{-1/2}$.

Other results about convergence time and conditions regarding the binary Voter Model and the network underlying topology can be found in the works of (COX, 1989; SOOD; REDNER, 2005; LIGGETT, 2005). More recent applications of the Voter Model for distributed decision-making process and discrete opinion formation are pointed out in (YILDIZ et al., 2011; FOTOUHI; RABBAT, 2014; VALENTINI et al., 2014; REDNER, 2019). Chapter 4 presents a variation of the Voter Model described by Equation (2.8), in a way that it can handle up to n unique discrete-space information at once, describing their occurrence probability individually for each node over time and also providing the consensus probability for every unique information of the system regarding its underlying network topology.

Example 2.3.2. Assume there is an interconnected system whose network topology is defined by the digraph of Figure 2.1c. All nodes have a unique discrete-space information at the beginning, in a way that the network state at that time is $\boldsymbol{x}[0] = [\text{A},\ \text{B},\ \text{C},\ \text{D}]$ with the letters representing each discrete information. All nodes execute Equation (2.8) following the scheme of selection of one node with probability 1/4 for each time step. Figure 2.3 depicts the trajectory of the information state of each node over time. The consensus happens over information B at time step 9. In fact, as this algorithm is asynchronous, each time step k can be understood as a random interaction between nodes, so the consensus happens after 9 interactions.

2.4. Summary

This chapter has presented the underlying structures which form the networks and determine its diffusion capability over time. They are the key pieces to describe the symbolic world of interconnected systems, since a simple tie among living cells until the complex social interactions of humankind. The graphs are accurate mathematical representations of networks. Their components, such as vertices and edges, are like bricks that together can reproduce surprisingly complex systems. Besides that, the relationships between such components can drive an entire world of possibilities, allowing complex behaviours to emerge and evolve along the time. The basis of such behaviours is the interaction among the nodes that, together, form the networked system. This interaction is controlled by the so-called diffusion dynamics, the rules that drive the information

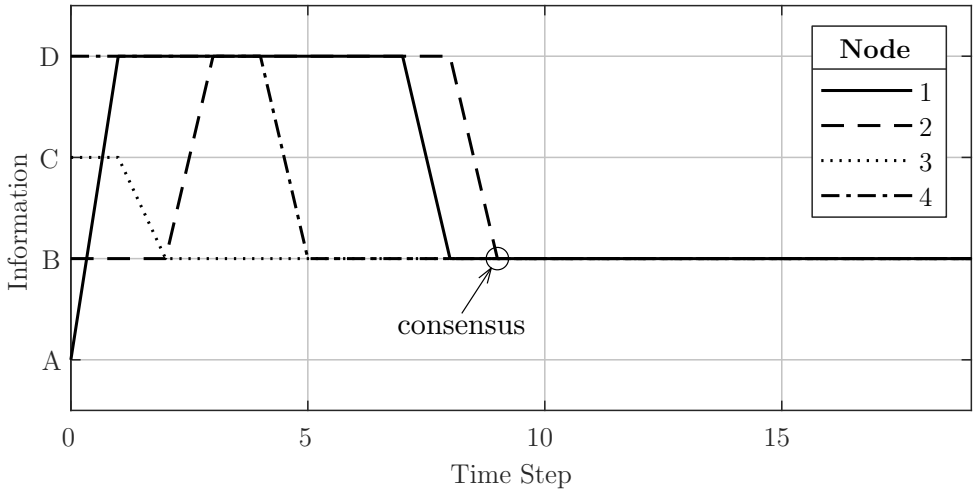


Figure 2.3: Information state trajectory of the voter model over time for each node.

exchanging in the entire system. When such dynamics are cohesive, an emergent behaviour called consensus appears, allowing the individuals of the networked system to cooperate. This feature turns networks into a mighty tool to solve huge problems in a distributed and autonomous way.

Chapter 3

Network Topology Control for Connectivity Maintenance

The communication topology is essential to allow cooperation among individuals of an interconnected system since they must exchange information and reach an agreement over a particular parameter of interest. The network topology must be connected to ensure all agents can exchange information and make decisions that consider everyone in the group. Through the changing of the communication topology, it is possible to ensure fault-tolerant topologies that still connected even with the failure of some of its nodes and links. When the network nodes are mobile robots, such topology manipulation can be done through their appropriated position displacement performed by approaches of topology control for connectivity maintenance.

This chapter extends an approach of topology control for connectivity maintenance, presented in (CARVALHO et al., 2015; CARVALHO; MORENO, 2015; CARVALHO, 2015), to deal with the communication signal strength variation and collision avoidance. As a minor contribution of this thesis, it keeps the same formulation and scenario of the former work, considering a group of planar mobile robots with single integrator dynamics that have to maximise the sensed (or covered) area of the team, ensuring fault tolerance to eventual disconnections and avoiding collision between them. It is done by building a kind of fault-tolerant communication topology derived from the graph concept of vertex connectivity (cf. Chapter 2), and by reducing the sensing area overlap of each robot, respecting some constraints on their minimum distance. An optimisation approach for non-convex problems applied to the distributed connectivity control ensures collision avoidance during the performed cooperative task.

The proposed approach is composed of two parts. Firstly, the Travel Salesman Problem (TSP) is solved to get the Hamiltonian cycle that is used to fix the vertex criticality. Secondly, a consensus algorithm is used within a Model Predictive Control (MPC) framework to move the robots according to the TSP's solution, while maximising the team sensing coverage area and respecting a safety area for each robot.

3.1. System Model

The considered system is fully distributed. That means the information in each robot is composed only of the information of its immediate neighbours and from their neighbours. It is called 2-hop neighbourhood, and is defined according to Equation (2.1), resulting in:

$$\mathcal{N}_i^2 = \{j \in \mathcal{N}_i; q \in \mathcal{V} \backslash \mathcal{N}_i : (j, q) \in \mathcal{E}\}.$$

There are many ways to determine when two robots are neighbours. This work uses

the *Received Signal Strength Indicator* (RSSI), represented by w_{ij} for each robot pair i and j. It returns the received power associated with the reception of a message and may allow determining how close a link is from disruption. The neighbourhood relationship for a robot i is defined as follows:

$$a_{ij}(t) = \begin{cases} 1 & \text{if } w_{ij}(t) \geq r_i^{\text{rssi}} \\ 0 & \text{if } w_{ij}(t) < r_i^{\text{rssi}}, \end{cases}$$

where r_i^{rssi} is the communication threshold given by the minimum readable RSSI level in node i, i.e. it cannot read any information if the signal from its sender is weaker than r_i^{rssi}. Note that the RSSI readings w_{ij} for each robot pair i and j vary with time, resulting in a time-varying topology.

It is also possible to estimate the link length (distance) as a function of propagation loss according to a model such as *Log-Normal Shadow Model* (ADEWUMI et al., 2013), which is suitable for wireless networks in free space. Thus, the relationship of this model with a Euclidean-based model is dictated by the following equation:

$$w_{ij} = -10\phi \log_{10}(d_{ij}) + C, \tag{3.1}$$

where d_{ij} is the euclidean distance between the robots i and j, ϕ is an environment-dependent path-loss exponent (for free space $\phi \approx 2$) and C is a fixed constant that is used to compensate model uncertainties.

Using the Euclidean model, each robot i has a communication radius (r_i^{com}) that represents the maximum transmission distance and a coverage radius (r_i^{cov}) that is its maximum sensing distance. As can be seen in Figure 3.1, this model is less accurate than the one based on RSSI. Generally, in real world applications, the RSSI is time varying and barely depends on the distance between the nodes. Environment noises are more significant in the RSSI readings of a network, and such aspects cannot be sensed by approaches based only on Euclidean model.

Also, each robot has a safety area around it, represented by the small circles in Figure 3.1. The term r_i^{min} represents the minimum distance a robot i can be near to any other without collision. Such a constraint is considered during the position displacement for all robots. With that, their minimum distance is indeed never violated during the cooperative task, as the next sections show.

3.2. Increasing Graph Connectivity with Virtual Links

The graph connectivity is increased by a method based on the solution of the Travelling Salesman Problem (TSP). It is used because the solution of TSP always results in a minimum bi-connected graph, as stated in the work of Frederickson and Ja'Ja' (1982), and that is the minimum fault-tolerant topology a network can have.

The TSP problem is built and solved individually by each robot, in a distributed way using 2-hop neighbourhood information. Such a solution is obtained from the classical integer linear programming formulation, proposed by Miller et al. (1960), adapted to work in a decentralised way using RSSI readings instead of Euclidean positions. The

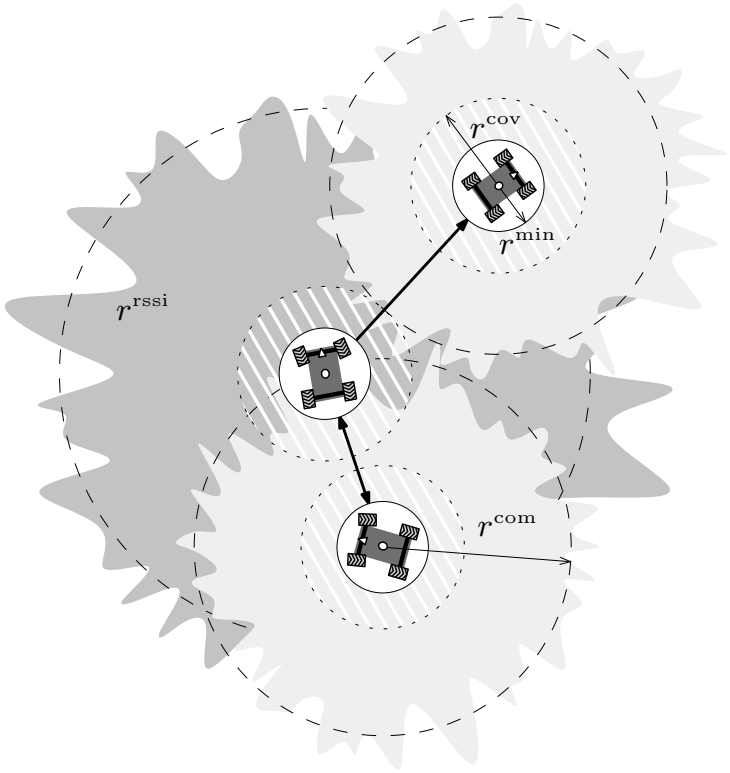


Figure 3.1: Communication model defined by the distance between robots. Bold arrows represent the communication links, the small circles are the safety area of each robot, the dotted circles are the sensing area, the dashed circles are the communication radii, and the grey area around each robot represents its communication signal.

formulation is defined for each robot i as follows:

$$
\begin{aligned}
\min_{\substack{\boldsymbol{\mathcal{H}}_i \in \mathbb{R}^{n\times n},\\ \boldsymbol{o}\in\mathbb{Z}^n}} \quad & \sum_{j=1}^{n}\sum_{q=1}^{n} |w_{jq}|\alpha_{jq}\varphi_{ijq} & & \\
\text{s.t.} \quad & \sum_{j=1}^{n} \alpha_{jq}\varphi_{ijq} = 1 & & j \neq q \quad q = 1,\dots,n, \\
& \sum_{q=1}^{n} \alpha_{jq}\varphi_{ijq} = 1 & & q \neq j \quad j = 1,\dots,n, \\
& o_j - o_q + n\alpha_{jq}\varphi_{ijq} \leq n-1 & & 2 \leq j \neq q \leq n, \\
& \varphi_{ijq} = a_{jq}\Big[a_{ij} + (1-a_{ij})a_{iq}\Big] & & 1 \leq j \neq q \leq n
\end{aligned}
\tag{3.2}
$$

where $w_{jq} \in \mathbb{R}$ is the RSSI reading of the link between j and q in dB, $\alpha_{jq} \in \{0,1\}$ is an integer variable that indicates the presence of a link between j and q, $\varphi_{ijq} \in \{0,1\}$ is an indicator function that is 1 when nodes j or q are 2-hop neighbours of i and 0 otherwise, $\boldsymbol{\mathcal{H}}_i = [\alpha_{jq}] \in \mathbb{R}^{n\times n}$ is the Hamiltonian matrix that has the TSP solution, and o_j is the j-th element of $\boldsymbol{o} \in \mathbb{Z}^n$, a free variable array. Note that, $\boldsymbol{\mathcal{H}}_i$ is different for each node i, since they use only 2-hop information to solve the optimisation program.

The Hamiltonian cycle derived from the TSP solution is the minimum bi-connected graph in the 2-hop neighbourhood of each robot. If the TSP input uses global network information, then the Hamiltonian cycle is the minimum bi-connected graph for the whole network (FREDERICKSON; JA'JA', 1982). However, the union of all Hamiltonian cycles

for all nodes contains the global optimum Hamiltonian cycle that is the global TSP problem's solution, i.e.

$$\boldsymbol{\mathcal{H}}^{\star} \subseteq \boldsymbol{\mathcal{H}}_1 \cup \boldsymbol{\mathcal{H}}_2 \cup \ldots \cup \boldsymbol{\mathcal{H}}_n.$$

Nevertheless, the obtained network may not be bi-connected since some links in the Hamiltonian cycle may not be feasible due to communication range violation. These are called *virtual links*, and they represent links that must be created by moving the robots.

3.3. Connectivity Control

To effectively fix the cut vertices' criticality, the robots must move towards each other, turning their virtual links derived from the TSP's solution into real ones. A consensus-based algorithm performs such a task in a decentralised way. This algorithm is formulated as a Model Predictive Controller (MPC), and it consists of a quadratic objective function J_i subject to some convex constraints, similar to the work of Ordoñez et al. (2012).

The objective function is defined as follows, to minimise the error between the robots' positions and the desired positions calculated according to the team goal, while it maximises the general covered area of the team:

$$\begin{aligned} J_i[k+l] = &\sum_{j=1}^{n} \|\boldsymbol{x}_i[k+l] - \boldsymbol{x}^*_{ij}[k]\|^2_{\gamma_i^x \psi_{ij}} \\ &+ \sum_{j=1}^{n} \|\boldsymbol{u}_i[k+l] + \boldsymbol{u}_j[k]\|^2_{\gamma_i^\alpha \varphi_{ij}} \\ &+ \sum_{j=1}^{n} \|\boldsymbol{u}_i[k+l] - \boldsymbol{u}_j[k]\|^2_{\gamma_i^u (\psi_{ij} - \alpha_{ij})} \\ &+ \|\boldsymbol{\Delta u}_i[k+l]\|^2_{\gamma_i^\Delta} \end{aligned} \tag{3.3}$$

where $\|\boldsymbol{x}\|^2_{\boldsymbol{Q}} \equiv \boldsymbol{x}^\mathsf{T}\boldsymbol{Q}\boldsymbol{x}$ is the quadratic norm of $\boldsymbol{x}$ induced by a matrix[1] $\boldsymbol{Q}$, l is the prediction instant, k is the current discrete time index, $\boldsymbol{x} \in \mathbb{R}^2$ is the state array for each robot with the coordinates in x-axis, and y-axis; $\boldsymbol{u} \in \mathbb{R}^2$ is the control signal array for each coordinate, $\gamma_i^x, \gamma_i^\alpha, \gamma_i^u, \gamma_i^\Delta \in \mathbb{R}$ are weight constants[2] to calibrate the control behaviour for each robot.

The variables ψ_{ij} and φ_{ij} are activation functions that enable some part of the objective function according to the nature of the link between robots i and j, they are defined as:

$$\psi_{ij} = a_{ij} + \varphi_{ij} \qquad \varphi_{ij} = (1 - a_{ij})\alpha_{ij}.$$

The first term of (3.3) is used to find the position of robot i that minimizes the distance to each neighbour j ($\boldsymbol{x}^*_{ij}$), to keep the robots together, while forcing their relative distances to be as close as possible to the sum of respective sensing coverage ranges. Consequently, it tries bringing the sensing areas together without overlapping, thus maximizing the total covered area. That position, to x and y coordinates, is defined as follows:

$$\boldsymbol{x}^*_{ij} = \frac{1}{d_{ij}}(\boldsymbol{x}_i - \boldsymbol{x}_j)(r_i^{\text{cov}} + r_j^{\text{cov}}) + \boldsymbol{x}_j. \tag{3.4}$$

[1] The matrix $\boldsymbol{Q}$ must be symmetric positive definite, and in this case $\boldsymbol{Q} \in \mathbb{R}$, i.e. it is a scalar.

[2] They are chosen regarding the application scenario and for general cases the following inequality holds: $\gamma_i^x > \gamma_i^\alpha > \gamma_i^u > \gamma_i^\Delta > 0$.

To adapt such solution to be sensitive to RSSI variations, the approximated position error given from the RSSI Log-Normal Shadow Model in Equation (3.1) replaces the inverse of the Euclidean distance between the robots i and j in Equation (3.4), resulting in

$$\boldsymbol{x}_{ij}^{*} = 10^{(w_{ij}-C)/(10\phi)}(\boldsymbol{x}_i - \boldsymbol{x}_j)(r_i^{\text{cov}} + r_j^{\text{cov}}) + \boldsymbol{x}_j, \tag{3.5}$$

where $w_{ij} \in \mathbb{R}$ is the RSSI reading for the link between the robots i and j. This function is used to amplify the position error according to the actual link conditions given by the RSSI readings.

Figure 3.2 depicts the optimal position to maximise the coverage area without overlapping. Note that, the Euclidean distance d_{ij} is approximated using RSSI readings of the link between robots i and j.

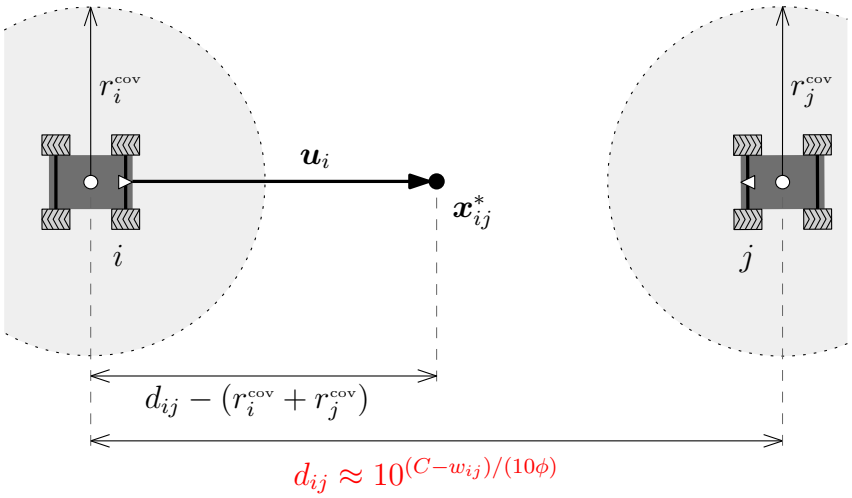


Figure 3.2: Allocated position to robot i considering its neighbour j's position.

The second term of (3.3) moves the virtual neighbours towards each other to make their link eventually become a real link. The activation function φ_{ij} is different from zero when the robots i and j are virtual neighbours. The third term is a flocking behaviour term that approximates the robots velocities making the robots move as a group. This term is enabled only when the robots i and j are real neighbours. Finally, the fourth term is the control effort penalization, and it is used to smooth the output control signal.

The whole distributed MPC program for each robot i is defined as follows:

$$\begin{aligned}
\min_{\boldsymbol{u}_i \in \mathbb{R}^2} \quad & \sum_{l=1}^{p} J_i[k+l] \\
\text{s.t.} \quad & \boldsymbol{x}_i[k+l] = \boldsymbol{x}_i[k+l-1] + h\boldsymbol{u}_i[k+l] && l = 1, \ldots, p, \\
& \boldsymbol{\Delta u}_i[k+l] = \boldsymbol{u}_i[k+l] - \boldsymbol{u}_i[k+l-1] && l = 1, \ldots, p, \\
& \underline{u}_i \leq \boldsymbol{u}_i[k+l] \leq \overline{u}_i && l = 1, \ldots, p, \\
& \|\boldsymbol{x}_i[k+l] - \boldsymbol{x}_j[k]\|_{a_{ij}} \leq r_j^{\text{com}} && l = 1, \ldots, p \quad j = 1, \ldots, n
\end{aligned} \tag{3.6}$$

where p is the prediction horizon, h is the sampling time, $\underline{u}_i \in \mathbb{R}$ and $\overline{u}_i \in \mathbb{R}$ are the saturation of the control signals.

Program (3.6)'s solution is the optimal control output for p predicted instants. However, only the first prediction is used to move the robot, like in a conventional MPC approach. For more information about MPC theory and optimisation, the interested reader is referred to (CAMACHO; ALBA, 2007) and (CAMPONOGARA et al., 2002).

3.3.1. Collision Avoidance

Note that, Equation (3.6) deals with the constraint to avoid communication range violation, but it does not consider any collision avoidance requirement. That happens because the collision avoidance constraint assumes the form

$$\left(\|\boldsymbol{x}_i[k+1] - \boldsymbol{x}_j[k]\| - r_i^{\min} - r_j^{\min}\right) a_{ij} \geq 0 \quad j = 1, \ldots, n \tag{3.7}$$

which is not convex and cannot be solved directly by the optimisation Program (3.6).

To overcome such a limitation, one can use a technique called Sequential Convex Programming (SCP) that is a local optimisation approach for non-convex problems based on solving a series of convex sub-problems. The main idea of the SCP is to find a good enough solution in a region around an initial feasible point of a convex approximation of the original non-convex problem. The SCP application origins are found in (SVANBERG, 1987), also followed by (ZILLOBER et al., 2004), for mechanical and structural problems, where it is used first-order approximation on functions limited by moving asymptotes to the operation domain, instead of complex functions whose convexity is not guaranteed.

The main idea is to find a convex approximation to the collision avoidance constraint represented by Equation (3.7), then use that in the original optimisation problem to find a solution that is valid at least for a small region (called trust region) of the search space in the problem with the original constraint. By doing it iteratively, starting always from the last feasible solution, the SCP can solve the original non-convex optimisation problem.

In this case the trust region is a box around the entry point $\hat{\boldsymbol{u}}_i^s \in \mathbb{R}^2$ obtained from the last feasible generated control action for the SCP iteration s, defining the domain of feasible convex set in which the next control actions must be

$$\mathfrak{U}^s = \{\boldsymbol{u}_i[k+1] \in \mathbb{R}^2 : \|\boldsymbol{u}_i[k+1] - \hat{\boldsymbol{u}}_i^s\|_1 \leq \rho_i\}. \tag{3.8}$$

where $\boldsymbol{u}_i[k+1]$ is the first generated MPC's control action for the current SCP iteration, $\|\cdot\|_1$ is the ℓ_1-norm and $\rho_i \in \mathbb{R}$ is the radius of the search region.

The first step is performed by squaring Inequality (3.7) and by replacing $\boldsymbol{x}_i[k+1]$ regarding the generated control action, yielding

$$c_{ij}(\boldsymbol{u}_i[k+1]) = \left((r_i^{\min} + r_j^{\min})^2 - \|h\boldsymbol{u}_i[k+1] + \boldsymbol{x}_i[k] - \boldsymbol{x}_j[k]\|^2\right) a_{ij} \leq 0$$

then the first order Taylor series is applied over it, producing the following affine approximation

$$\begin{aligned} \tilde{c}_{ij}(\boldsymbol{u}_i[k+1], \hat{\boldsymbol{u}}_i^s) &= c_{ij}(\hat{\boldsymbol{u}}_i^s) + \boldsymbol{\nabla} \boldsymbol{c}_{ij}(\hat{\boldsymbol{u}}_i^s)^{\mathsf{T}}(\boldsymbol{u}_i[k+1] - \hat{\boldsymbol{u}}_i^s) \leq 0 \qquad && j = 1, \ldots, n \\ \boldsymbol{\nabla} \boldsymbol{c}_{ij}(\hat{\boldsymbol{u}}_i^s) &= \Big(- 2h^2 \hat{\boldsymbol{u}}_i^s - 2h\boldsymbol{x}_i[k] + 2h\boldsymbol{x}_j[k]\Big) a_{ij} && j = 1, \ldots, n \end{aligned}$$

where $\hat{\boldsymbol{u}}_i^s$ is the first generated control signal, $\boldsymbol{u}_i[k+1]$, got at the previous SCP iteration. The approximated constraint $\tilde{c}_{ij}$ is valid only for each SCP iteration s.

By running iteratively the Program (3.6) for a given time index k, using the approximated constraint updated for each new execution of the SCP, results in a control signal that is optimal to the original problem and that does not violate the original non-convex constraint expressed in Equation (3.7). This optimised control signal is found when its difference between two consecutive SCP iterations is lower than a threshold, i.e. until $\|\hat{\boldsymbol{u}}_i^s - \hat{\boldsymbol{u}}_i^{s+1}\| \leq \epsilon$, where $\epsilon \in \mathbb{R}$ is a tolerance value.

The geometric interpretation of the linearisation is shown in Figure 3.3 by two robots i and j. The robot i shall avoid the region delimited by the circle defined by the function

c_{ij}. It is clear that the complement of the dark set is a non-convex set. The "safety region" $\mathfrak{U}^s$ is defined regarding Equation (3.8), and it represents the search space for the optimal control outputs that do not violate the non-convex constraints at SCP iteration s. Such area determines feasible positions where the robot i can take place without collision with its nearest neighbour.

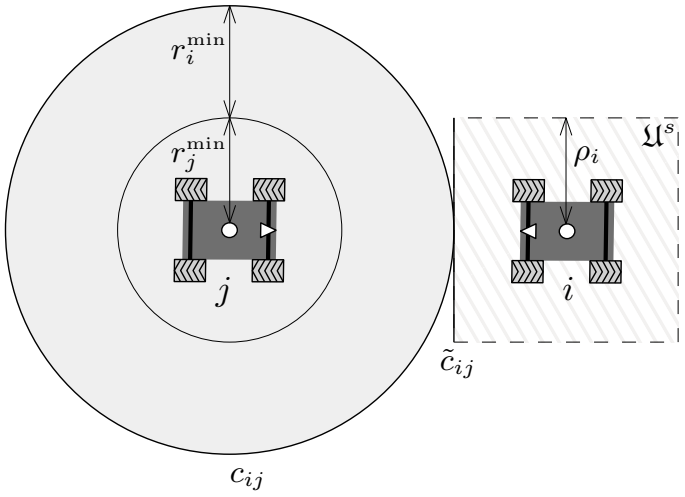


Figure 3.3: Geometric interpretation of the SCP linearisation from robot i perspective: the circle c_{ij} defines the limits of the avoided area provided by the original non-convex constraint, $\tilde{c}_{ij}$ is the representation of the convex constraint for each SCP iteration, $\mathfrak{U}^s$ is the search area of one iteration s of the SCP algorithm.

3.3.2. Matrix Form

Program (3.6) must be written as a quadratic problem in canonical form, with the shape $\frac{1}{2}\boldsymbol{x}^\mathsf{T}\boldsymbol{H}\boldsymbol{x} + \boldsymbol{g}^\mathsf{T}\boldsymbol{x}$, where $\boldsymbol{H}$ is a positive semi-definite Hessian matrix, and $\boldsymbol{g}$ is a gradient array since a commercial solver is used on it.

First, it is defined the dynamics of robot i for all p predicted instants, in the following way:

$$\underbrace{\begin{bmatrix}\boldsymbol{x}_i[k+1]\\ \boldsymbol{x}_i[k+2]\\ \vdots\\ \boldsymbol{x}_i[k+p]\end{bmatrix}}_{\hat{\boldsymbol{X}}_i} = \underbrace{\begin{bmatrix}\boldsymbol{x}_i[k]\\ \boldsymbol{x}_i[k]\\ \vdots\\ \boldsymbol{x}_i[k]\end{bmatrix}}_{\boldsymbol{X}_i} + h\underbrace{\begin{bmatrix}\boldsymbol{I}_2 & \boldsymbol{0}_{2\times2} & \dots & \boldsymbol{0}_{2\times2}\\ \boldsymbol{I}_2 & \boldsymbol{I}_2 & \dots & \boldsymbol{0}_{2\times2}\\ \vdots & \vdots & \ddots & \vdots\\ \boldsymbol{I}_2 & \boldsymbol{I}_2 & \dots & \boldsymbol{I}_2\end{bmatrix}}_{\boldsymbol{T}_x}\underbrace{\begin{bmatrix}\boldsymbol{u}_i[k+1]\\ \boldsymbol{u}_i[k+2]\\ \vdots\\ \boldsymbol{u}_i[k+p]\end{bmatrix}}_{\hat{\boldsymbol{U}}_i}$$

then the allocated position and neighbours' position and control arrays are written as

$$\boldsymbol{X}_{ij}^* = \begin{bmatrix}\boldsymbol{x}_{ij}^*[k]\\ \boldsymbol{x}_{ij}^*[k]\\ \vdots\\ \boldsymbol{x}_{ij}^*[k]\end{bmatrix} \qquad \boldsymbol{X}_j = \begin{bmatrix}\boldsymbol{x}_j[k]\\ \boldsymbol{x}_j[k]\\ \vdots\\ \boldsymbol{x}_j[k]\end{bmatrix} \qquad \boldsymbol{U}_j = \begin{bmatrix}\boldsymbol{u}_j[k]\\ \boldsymbol{u}_j[k]\\ \vdots\\ \boldsymbol{u}_j[k]\end{bmatrix}$$

and the control signal variation is written as

$$\underbrace{\begin{bmatrix}\boldsymbol{\Delta u}_i[k+1]\\ \boldsymbol{\Delta u}_i[k+2]\\ \vdots\\ \boldsymbol{\Delta u}_i[k+p]\end{bmatrix}}_{\boldsymbol{\Delta U}_i} = \underbrace{\begin{bmatrix}\boldsymbol{I}_2 & \boldsymbol{0}_{2\times2} & \dots & \boldsymbol{0}_{2\times2}\\ -\boldsymbol{I}_2 & \boldsymbol{I}_2 & \dots & \boldsymbol{0}_{2\times2}\\ \vdots & \vdots & \ddots & \vdots\\ \boldsymbol{0}_{2\times2} & \boldsymbol{0}_{2\times2} & \dots & \boldsymbol{I}_2\end{bmatrix}}_{\boldsymbol{T}_\Delta = h\boldsymbol{T}_x^{-1}} \hat{\boldsymbol{U}}_i - \underbrace{\begin{bmatrix}\boldsymbol{u}_i[k]\\ 0\\ \vdots\\ 0\end{bmatrix}}_{\boldsymbol{U}_{i,0}},$$

where $\boldsymbol{x}_i, \boldsymbol{u}_i \in \mathbb{R}^2$ contains, respectively, the values of robot i's position and control signals to the axis x and y; $\boldsymbol{I}_2$ is an identity matrix with dimension 2; and $\boldsymbol{0}_{2\times2}$ is a square matrix of zeros with dimension 2.

So, rewriting the Equation (3.3) in a vectorial way results:

$$\begin{aligned} J_i = &\sum_{j=1}^{n} \|\boldsymbol{X}_i + \boldsymbol{T}_x\hat{\boldsymbol{U}}_i - \boldsymbol{X}_{ij}^*\|^2_{\gamma_i^x\psi_{ij}}\\ &+ \sum_{j=1}^{n} \|\hat{\boldsymbol{U}}_i + \boldsymbol{U}_j\|^2_{\gamma_i^\alpha\varphi_{ij}}\\ &+ \sum_{j=1}^{n} \|\hat{\boldsymbol{U}}_i - \boldsymbol{U}_j\|^2_{\gamma_i^u(\psi_{ij}-\alpha_{ij})}\\ &+ \|\boldsymbol{T}_\Delta\hat{\boldsymbol{U}}_i - \boldsymbol{U}_{i,0}\|^2_{\gamma_i^\Delta} \end{aligned} \tag{3.9}$$

Finally, by rewriting Equation (3.9) isolating the terms $\hat{\boldsymbol{U}}_i^\mathsf{T}\hat{\boldsymbol{U}}_i$ and $2\hat{\boldsymbol{U}}_i$, and by adding the collision avoidance constraint, the distributed MPC optimisation program described by Equation (3.6) can be represented in its compact canonical form for each node i at the SCP step s and discrete time index k as

$$\begin{aligned} \Upsilon_i(\hat{\boldsymbol{u}}_i^s, s): \min_{\hat{\boldsymbol{U}}_i\in\mathbb{R}^{2p}} \quad & \frac{1}{2}\hat{\boldsymbol{U}}_i^\mathsf{T}\boldsymbol{H}_i\hat{\boldsymbol{U}}_i + \boldsymbol{g}_i\hat{\boldsymbol{U}}_i\\ \text{s.t.} \quad & -\hat{\boldsymbol{U}}_i + \boldsymbol{1}_{2p}\underline{u}_i \le \boldsymbol{0}_{2p},\\ & \hat{\boldsymbol{U}}_i - \boldsymbol{1}_{2p}\overline{u}_i \le \boldsymbol{0}_{2p},\\ & \|\boldsymbol{x}_i[k] + h\boldsymbol{u}_i[k+1] - \boldsymbol{x}_j[k]\|a_{ij} \le r_j^{\text{com}} \quad j=1,\dots,n,\\ & \tilde{c}_{ij}(\boldsymbol{u}_i[k+1], \hat{\boldsymbol{u}}_i^s) \le 0 \quad j=1,\dots,n,\\ & \|\boldsymbol{u}_i[k+1] - \hat{\boldsymbol{u}}_i^s\|_1 - \rho_i \le 0,\\ & \hat{\boldsymbol{u}}_i^{s+1} = \boldsymbol{u}_i[k+1] \end{aligned} \tag{3.10}$$

where

$$\boldsymbol{H}_i = \sum_{j=1}^{n}\left(\boldsymbol{T}_x^\mathsf{T}\gamma_i^x\psi_{ij}\boldsymbol{T}_x + \gamma_i^\alpha\varphi_{ij}\boldsymbol{I}_{2p} + \gamma_i^u(\psi_{ij}-\alpha_{ij})\boldsymbol{I}_{2p}\right) + \boldsymbol{T}_\Delta^\mathsf{T}\gamma_i^\Delta\boldsymbol{T}_\Delta$$

and

$$\boldsymbol{g}_i = \sum_{j=1}^{n}\left[\left(\boldsymbol{X}_i - \boldsymbol{X}_{ij}^*\right)^\mathsf{T}\gamma_i^x\psi_{ij}\boldsymbol{T}_x + \boldsymbol{U}_j^\mathsf{T}\gamma_i^\alpha\varphi_{ij} - \boldsymbol{U}_j^\mathsf{T}\gamma_i^u(\psi_{ij}-\alpha_{ij})\right] - \boldsymbol{U}_{i,0}^\mathsf{T}\gamma_i^\Delta\boldsymbol{T}_\Delta$$

where $\boldsymbol{I}_{2p}$ is an identity matrix with dimension $2p$; $\boldsymbol{0}_{2p}$ and $\boldsymbol{1}_{2p}$ are arrays of zeros and ones with order $2p$, respectively.

The generated trajectories are the important values in the optimisation process, given by the decision variable $\boldsymbol{u}_i[k+1]$ that defines the next individual movement of

every robot i. For each SCP step s, it is generated an optimal value of $\boldsymbol{u}_i[k+1]$ which is settled as the next SCP iteration's point, i.e. $\hat{\boldsymbol{u}}_i^{s+1} = \boldsymbol{u}_i[k+1]$. The difference between the optimal control action resultant of the MPC and the current SCP point cannot be greater than ρ_i, so the new optimal control action will always be in the safety zone $\mathfrak{U}^s$.

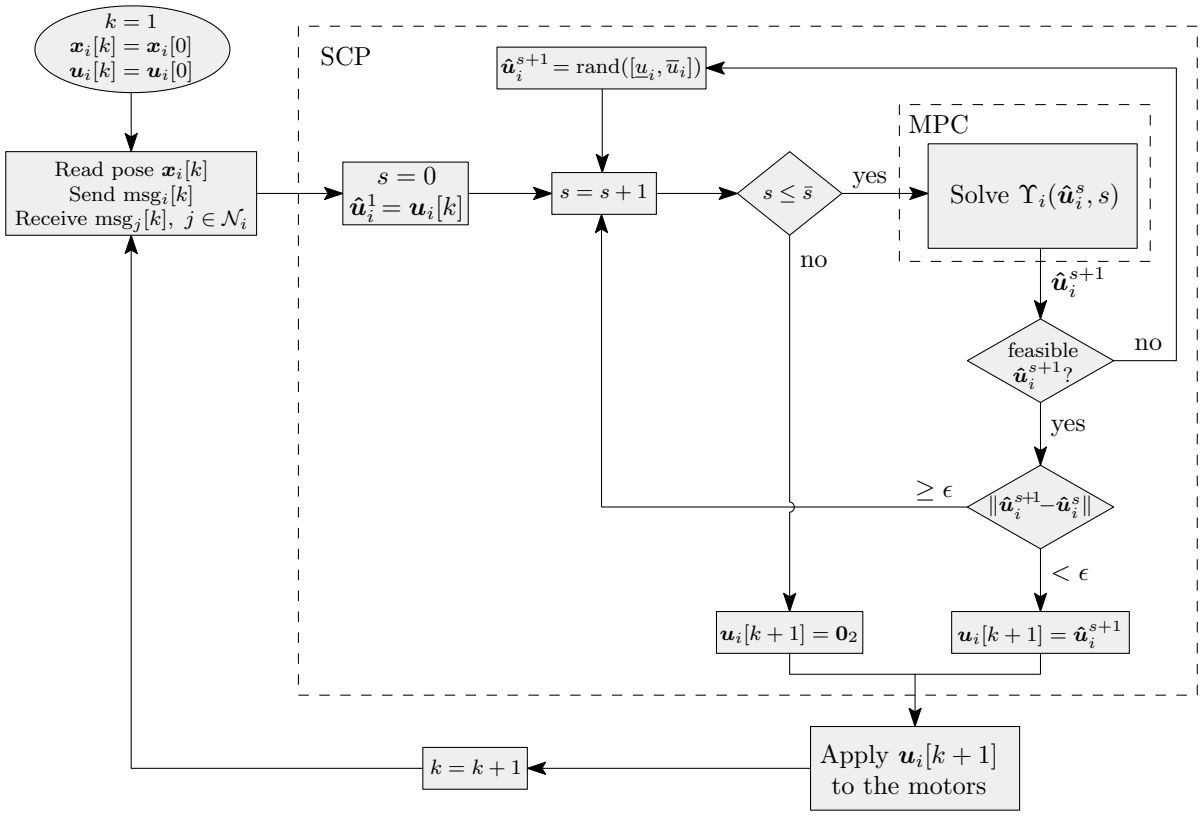


Figure 3.4: Execution of the SCP and MPC for a robot i: msg_i is a network message, $\text{rand}(\cdot)$ is a function that generates a uniformly distributed random value in the informed range, and $\bar{s}$ is the maximum number of SCP iterations.

Figure 3.4 depicts how the SCP is related to the MPC program, from robot i perspective. Note that, the MPC program, $\Upsilon_i(\boldsymbol{u}_i^s, s)$, is solved on each SCP iteration s, followed by the evaluation of the feasibility and convergence of the generated control signal. If such a value is not feasible in the original problem (e.g. it forces the violation of the collision avoidance constraint), then a random value is used as input to a new SCP iteration. When the convergence criterion is reached, the optimal generated control signal, $\hat{\boldsymbol{u}}_i^{s+1}$, (i.e., the first prediction of MPC generated at SCP iteration s) is sent to the motors, and the movement is performed. If the convergence criterion is not fulfilled, then a new SCP iteration starts (i.e. s increases in one) and a new execution of $\Upsilon_i(\boldsymbol{u}_i^s, s)$ is performed using the last generated control signal. The next discrete-time step starts with robot i reading its position and communicating with the others to get information about its neighbours, followed by the SCP iterations, and so forth. Note that, there might be a case where SCP procedure cannot find easily a solution that does not violate the constraints, keeping itself running for long periods. In this case, to avoid execution time problems, there is an upper bound $\bar{s} > 0$ to the maximum number of performed SCP iterations. Once this value is reached and no optimal control action is found, the robots receive $\mathbf{0}_n$ as its next control action, e.g. stay still until the next time step.

The complete topology control structure for a robot i is illustrated in Figure 3.5.

The COM block represents the communication layer with its appropriated protocols, the TSP block calculates the TSP solution and estimates the optimal position to maximises the coverage area of i, the variable $\boldsymbol{x}_j$ is the position array for each robot i's neighbour j, the SCP block represents the SCP algorithm within the MPC controller for the position displacement, and the R_i block represents the robot i dynamics.

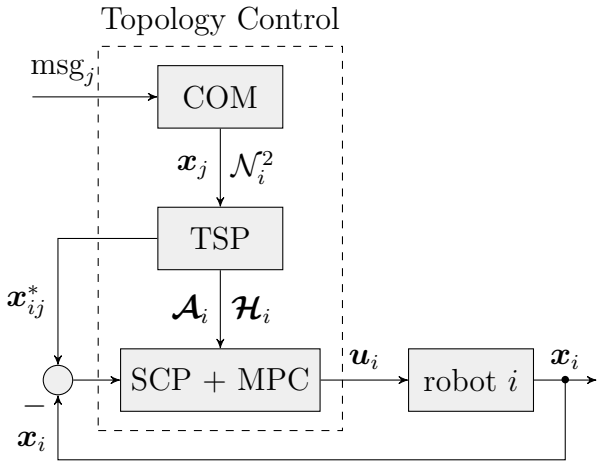


Figure 3.5: Topology control diagram.

This approach considers simplified holonomic robots, but it is also possible to contemplate nonholonomic robots just by adding a second motion control layer between the blocks SCP and R_i that "translates" the omnidirectional velocities into restricted ones. However, note that the proposed MPC controller does not take into account complex dynamics since only robots with single integrator dynamics are considered.

3.4. Experiments

In order to validate the effectiveness of the proposed method, some experiments with relevant operational scenarios are carried out. Three simulated experiments were performed: in the first one, the robots starting from an arbitrarily connected network must reach a bi-connected topology, then they have to recover this configuration after one robot drop out of the system. The second test aims to verify the capability of the proposed approach to sense the RSSI variations and adapt the network topology to them. The third test evaluates the collision avoidance capability of the proposed approach during the position displacement of all robots. All those tests are performed using planar and omnidirectional robots with the communication range of 10 meters, coverage radius of 4.5 meters, and safety radius of 1 meter, except in the third experiment, where the coverage range becomes zero for all robots.

Also, there is a practical experiment using two commercial robots from *Pioneer* with ROS[3] and four simulated ones. This experiment explores the case where two robots have to pass through a narrow path, composed by stationary robots, to reach a meeting point without colliding with them. The robots used have a differential drive, therefore, a *low-level* controller is used to "translate" the omnidirectional signals from the MPC to be compatible with them.

[3] <http://www.ros.org/>

The simulation setup is the same for all the experiments, and it is defined as follows:

$h = 0.01$ and 0.45	Integration step (s)
$p = 5$	MPC prediction horizon (steps)
$\gamma^x = 50$	Gain to the weight of the optimal position allocation
$\gamma^\alpha = 5$	Gain to the weight of the virtual links
$\gamma^u = \gamma^\Delta = 1$	Gains to the weight of control variation
$\overline{u} = 1$	Maximum linear velocity (m/s)
$\underline{u} = -1$	Minimum linear velocity (m/s)
$\rho = 1$	Radius of the SCP's search area
$\bar{s} = 30$	Maximum number of SCP's iterations
$\phi = 2$	Path loss exponent
$r^{\text{rssi}} = -23$	RSSI link threshold (dBm).

Note that the integration step of $h = 0.01$ was used in the simulated experiments (Sections 3.4.1 to 3.4.3), while the integration step of $h = 0.45$ was used with the real robots experiment (Section 3.4.4).

All experiments use Gurobi[4] as solver with the modeling package CVX[5] for building convex problems in Julia[6]. All the simulated robots were executed in an Intel i7 Quad-Core of 3.40GHz with 8GB of RAM. The commercial robots used are the *Pioneer* 3-AT and 3-DX, which have an onboard computer with an Intel i5 Quad-Core of 2GHz with 8GB of RAM each.

3.4.1. Bi-connecting an Arbitrary Network

In this simulation, the initial topology is not bi-connected, and it is composed of 10 robots, as can be seen in Figure 3.6a. After some iterations of the topology control approach, the network begins to increase its algebraic connectivity, as depicted in Figure 3.6b. The topology reaches bi-connectivity in Figure 3.6c. In Figure 3.6d, robot 3 drops out of the network, but it still a bi-connected network. Robot 1 drops out of the network in Figure 3.6e, turning it into a not bi-connected one (robot 5 is a critical node). Finally, in Figure 3.6f, the network bi-connectivity is restored through the creation of the links among robots 2, 4, and 6.

There is a time delay in the process of turn the network bi-connected, as can be seen in Figures 3.6a, 3.6b, and 3.6e. This time delay is proportional to the area where the robots are in and their maximum speed. In that period, the network has critical vertices and can be disconnected if such vertices are removed. For this reason, the proposed approach cannot ensure fault tolerance during all the task time. However, once the bi-connected topology is reached, that property is guaranteed even with the drop out of one robot.

3.4.2. Sensing Signal Strength Variation

In this experiment, the communication link between two robots is exposed to an external and intermittent attenuation caused by a signal interference or by the presence of obstacles in the environment. Such interference can broke the communication if the topology control approach is not sensitive to the RSSI variations.

[4] <http://www.gurobi.com/>
[5] <https://github.com/JuliaOpt/Convex.jl>
[6] <https://julialang.org/>

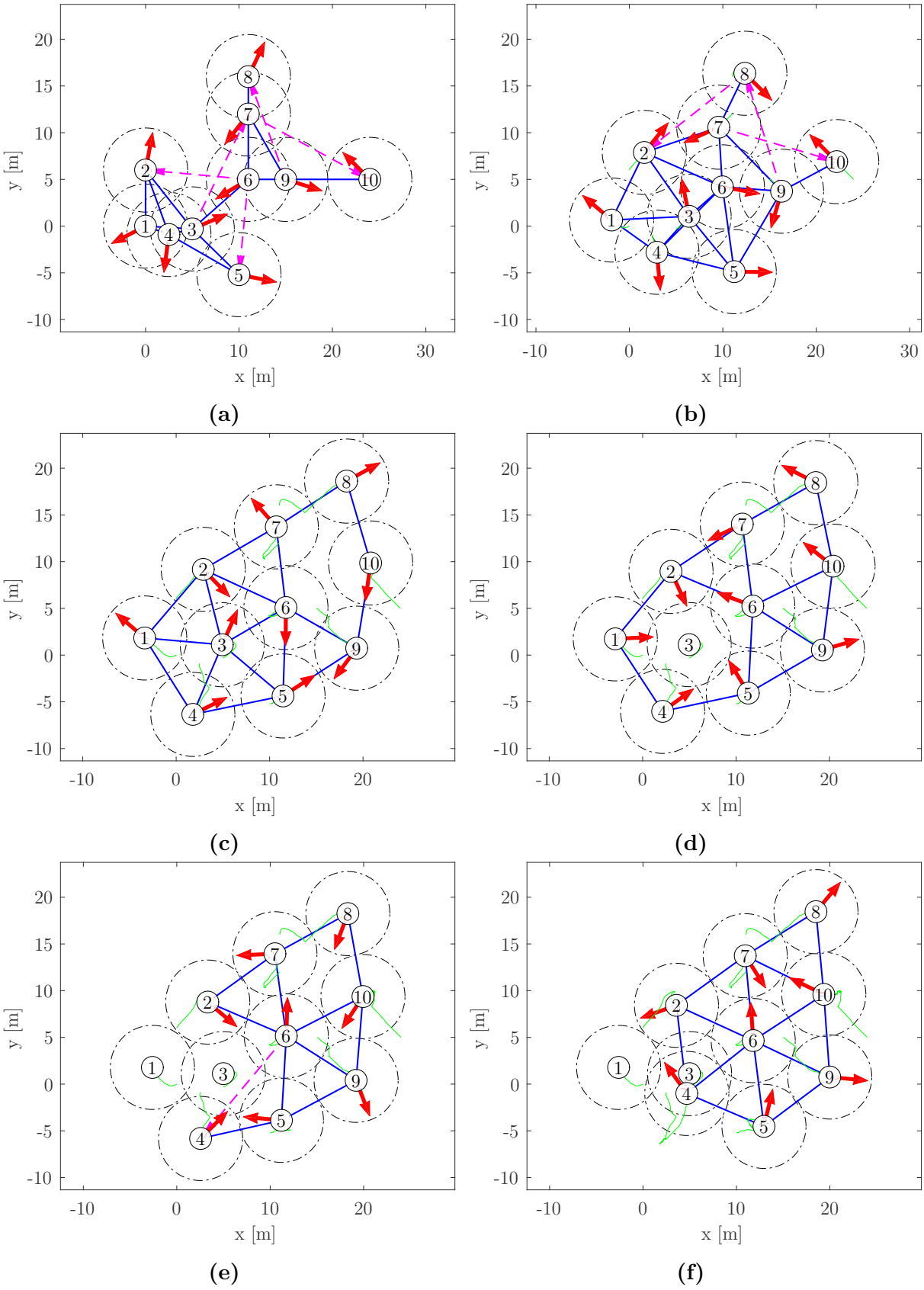


Figure 3.6: Bi-connecting a network. Circles are the sensing area of each robot, red arrows are the velocity vector, dashed lines are virtual links, continuous lines are regular links, and green lines are the path travelled by each robot.

As can be seen in Figure 3.7a and 3.7b, initially there is no interference in the link between the robots 1 and 2. After some time (near to 10 seconds of simulation), a step perturbation is inserted to the RSSI readings of their link, as shown in Figure 3.7d. Then, both robots reduce their distances to compensate the detected low communication signal, as depicted in Figure 3.7c. In Figure 3.7f the attenuation is removed from the RSSI readings at 25 seconds, approximately. With it, the robots can move back to their optimal positions where the coverage area is maximised, as can be seen in Figure 3.7e. Figure 3.7g shows the steady state of the system after the removal of the signal attenuation.

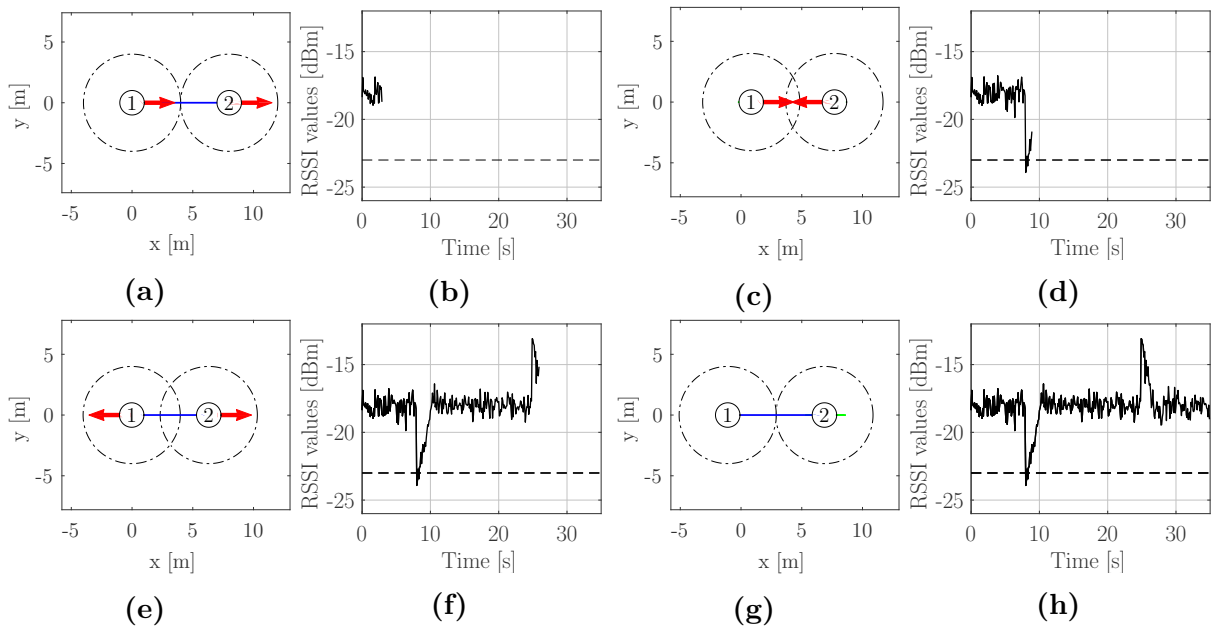


Figure 3.7: Link sensing to RSSI variation. Circles are the robots' coverage area, arrows are their velocity vector, and dashed lines in the RSSI plots are the communication threshold.

Note that, the optimal allocated position ($\boldsymbol{x}_{ij}^{*}$) of the robots changes according to the RSSI readings unlike in the pure Euclidean-based approach, in which such position is determined only by the coverage radius of each robot. Such strategy allows the system to sacrifice the coverage optimisation in favour of an increase in the link signal strength.

3.4.3. Evaluating the Collision Avoidance

In this experiment, the coverage range of each robot becomes equal to zero, in a way that the allocated position in the MPC algorithm for each neighbour is the same, forcing collision among them. The safety range is kept to be of 1 meter for all robots, and the collision avoidance constraint is enabled. The robots must bi-connect a network while avoiding collision with each other.

Figure 3.8 depicts the time evolution of the robots' positions, their communication links, as well as their collision areas. In Figure 3.8a, one can see that the initial topology is the same used in the first experiment, being not bi-connected. As shown in Figure 3.8b and 3.8c, after some iterations, the network becomes bi-connected. However, as can be seen in Figure 3.8d, the robots still trying to minimise their relative distance, since their coverage area is zero, resulting in a single allocated position for all robots that exchange information with each other. In Figures 3.8e and 3.8f, one can see that despite their

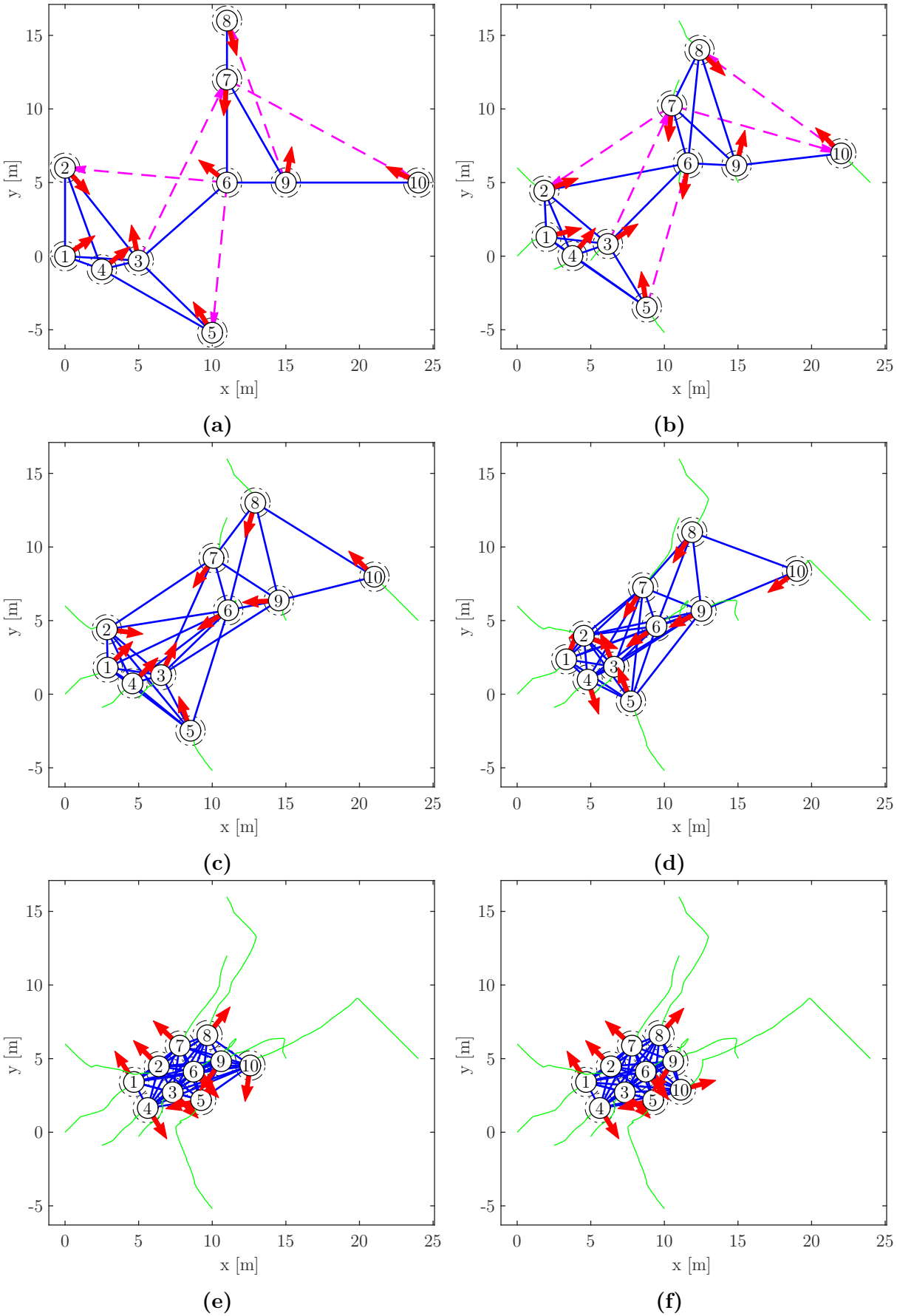


Figure 3.8: Bi-connecting a network with no coverage area. Circles are the safety area of each robot, red arrows are the velocity vector, dashed lines are virtual links, continuous lines are regular links, and green lines are the path travelled by each robot.

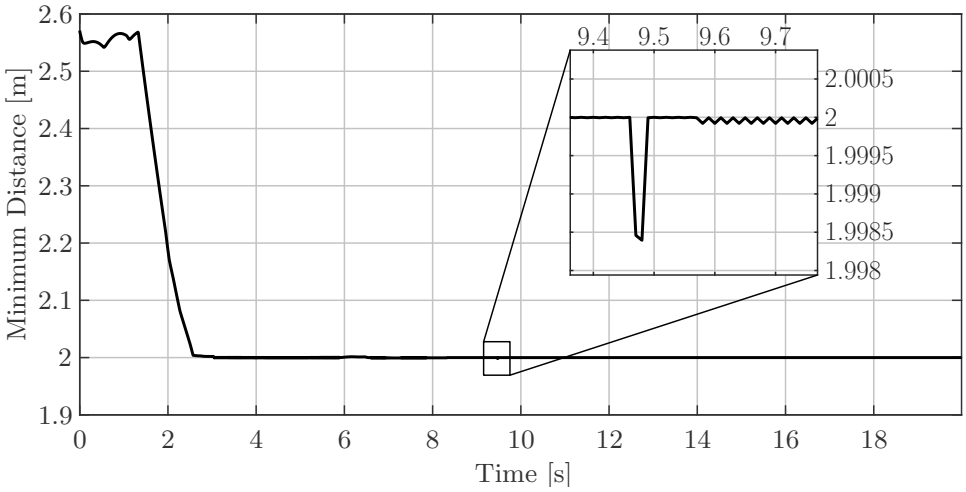


Figure 3.9: General minimum distance among all robots over time. Zoomed is a safety area violation.

searching for distance minimisation, the robots stop going towards each other when their safety areas are near to overlap, avoiding collision as intended by the collision avoidance constraint.

Figure 3.9 portrays the general minimum distance among all the robots. As one can see, their minimum distance decays fast (a little after two seconds of simulation time) reaching 2 meters, which is the minimum limit imposed by the sum of the safety radius of each robot. This value stays stable around 2 meters, but there are few violations in precise moments. The most relevant one happens between 9.4 and 9.5 seconds and is shown in the zoomed area of the figure. The violation is of 0.0015 meters, which means the robots performed an overlap of 0.3% in their safety areas. Despite small, such violation should not happen, since the collision avoidance constraints are imposed on the optimisation program. They are mainly caused by the linearisation performed over the former constraint, that takes into account the integration step size. Thus, the smaller is the integration step size, the smaller is the constraint violation, and ideally, if the integration step size is zero, there is no violation at all.

3.4.4. Collision Avoidance on Real Robots

This experiment aims to demonstrate the robots' behaviour when there is a narrow path between them and a reference point to evaluate the proposed collision avoidance strategy. It uses two commercial robots from *Pioneer* (robot 1 and 2) that must reach the reference point, and four simulated robots that are stationary and do not receive information from the real ones but send their position to create the narrow path. All robots have safety radius settled to 0.4 meters, and the *Pioneers* have communication radius of 5 meters. All coverage radius are settled to be equal to zero since this experiment is interested only on the evaluation of the collision avoidance approach.

It is performed 200 iterations of the proposed algorithm, using the integration step of 0.45 seconds, obtained from an empirical evaluation of the experiment scenario. Figure 3.10 shows the robots from Pioneer in the test field reaching the reference point, and the buckets representing the virtual stationary robots[7]. Figure 3.11 depicts the robots' traveled path

[7]Video available at <https://youtu.be/qY_oHi1HL8A>.

Figure 3.10: Robots from Pioneer performing the experiment.

to reach the reference and the stationary robots that compose the path's wall. As can be seen, their trajectory to the reference is free of collision with the stopped robots.

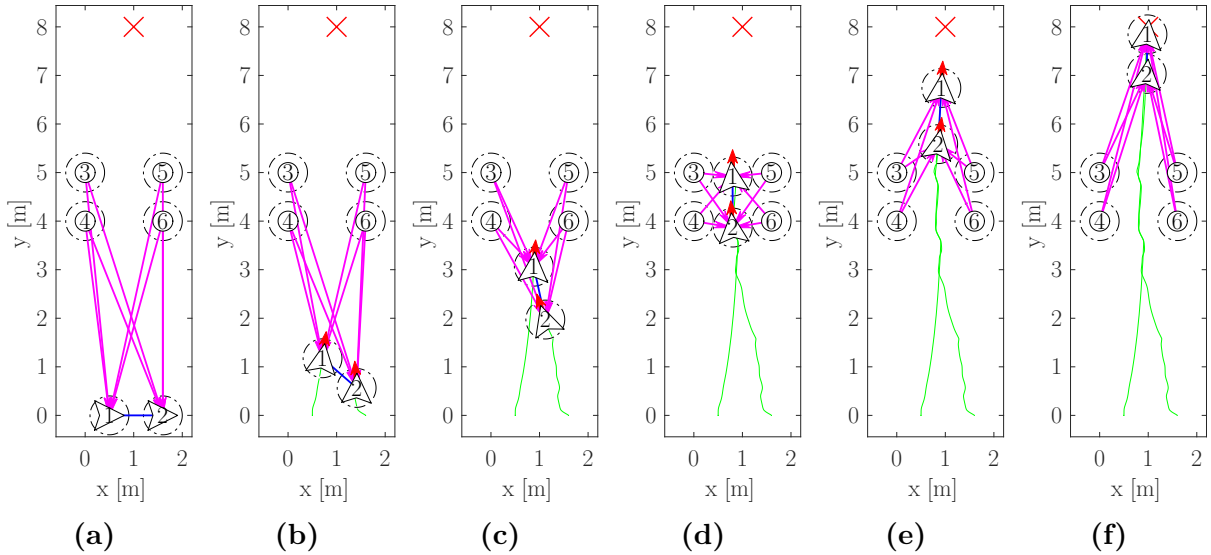


Figure 3.11: Collision avoidance on commercial robots. The dashed circles are the safety area of each robot, red arrows are the velocity vector, blue lines are regular links, magenta lines are directional links, and green lines are the path travelled by each robot.

By analysing Figure 3.12, one can see there are small violations of the safety areas. In this case, the most significant violation happens around 55 seconds of the experiment time, where the minimum distance between the robots reach 0.77 meters (a violation of 0.03 meters) which is not enough to result in a collision between them. Such violations mainly are caused by the approximation of the non-convex constraint, as stated before, besides that, in this case specifically, the robots have a noise associated with their odometry, which causes some imprecision in their position readings.

3.5. Summary

This chapter has presented the minor contribution of this thesis to the field of network topology control for connectivity maintenance in multi-robot systems: an approach that

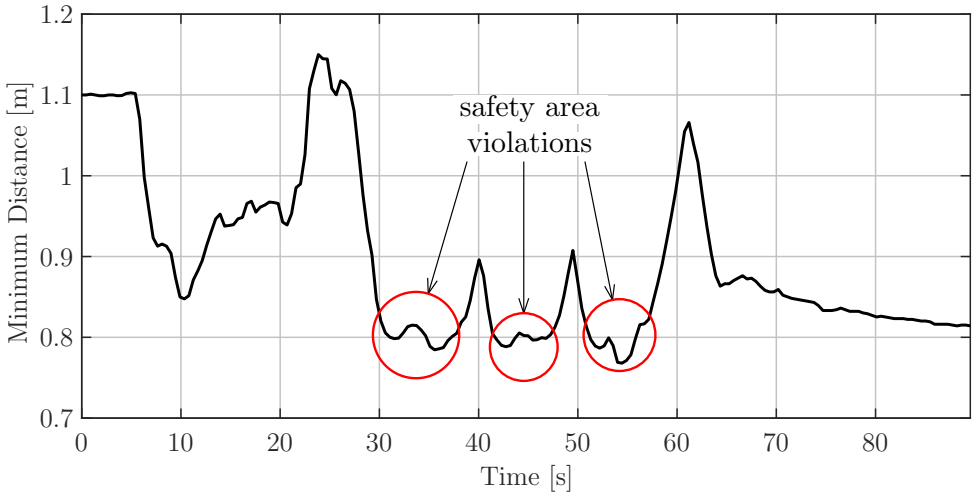


Figure 3.12: General minimum distance among all robots over time. Circles indicate safety area violations.

uses vertex-connectivity obtained from a distributed TSP problem to build a bi-connected network that is fault-tolerant. Such an approach is extended to deal with the signal strength readings of the links to improve the system sensibility to RSSI fluctuations that could break pure Euclidean-based approaches. Also, collision avoidance capabilities are added to the MPC-based framework used to displace the robots, performing connectivity control (turning the TSP's solution into a real topology configuration) and coverage control (minimising the robots' sensing area overlap). The experiments have shown that the proposed approach can turn any strongly connected network into a bi-connected one that is sensible to the variation of the communication signal in wireless networks and that, at the same time, avoids collision among all the robots of a multi-robot system in a decentralised way. Moreover, the last experiment demonstrates the viability of the proposed approach to be implemented in commercial robots to solve real-world problems.

Chapter 4

Network Design for Information Spreading Manipulation

As stated early, the topology has a great influence over information spreading in cooperative systems and it is one of the critical factors to determine the steady-state of consensus protocols in such systems. Thus, the manipulation of the network topology is a natural path to be followed by approaches that intend to control the information spreading over cooperative systems. To properly control the information spreading through the manipulation of the network topology, firstly, it is necessary to model the spreading process regarding the network fundamental components, such as links and vertices. Once this model is created, one can realise how to change the links and their weights in a way that the information spreading behaves as desired. The resultant designed topology can improve the agreement over a multi-agent system by ensuring a particular behaviour to its spreading dynamics.

This chapter presents an approach for network reweighting based on semidefinite programming in order to manipulate the convergence of a discrete-space consensus protocol in an interconnected system. It uses spectral optimisation for optimal weight design of the network adjacency matrix, in a way that the eigenvector centrality of this network (cf. Chapter 2) is changed, ensuring a certain behaviour to the underlying cohesive diffusion dynamics. The considered diffusion dynamics is a variation of the synchronous Voter Model for discrete-space information named here as Discrete-Space Random Consensus Protocol (DSRCP). Its dynamical spreading model is derived from concepts of epidemic diffusion and discrete-time Markov chains, allowing one to analyse the information spreading probabilities regarding the network topology and its influence in the consensus convergence over discrete information. The conditions under which this protocol reaches global consensus regarding topology requirements are pointed out. Also, it is shown that the steady-state of the spreading probabilities are fundamentally driven by the spectrum of the transition matrix of the proposed model, and those values are equivalent to the network eigenvector centrality which is controlled by the network reweighting approach.

4.1. System Spreading Model

This section presents the dynamical spreading model for the Discrete-Space Random Consensus Protocol (DSRCP), a distributed algorithm for agreement over countable information based on a synchronous version of the Voter Model, presented in Section 2.3.2. As a contribution of this thesis, this dynamical spreading model allows one to fully understand the information diffusion of the DSRCP over time regarding the network topology and its components.

4.1.1. Problem Statement

Assume there is a group of n autonomous agents trying to solve a decision-making problem cooperatively. Each agent knows only about their nearest neighbours, and there is no global information about the state of the network. It is also considered that the communication topology is time-invariant during the decision-making process. They must agree in a decentralised way about the value of a discrete parameter, which represents a decision over a set of values (e.g. meeting points). They shall use a type of discrete-space consensus protocol to solve such a problem since the parameter they want to estimate distributively belongs to a countable (and maybe symbolic) domain.

A possible approach to solve such a problem could be the Voter Model introduced in Section 2.3.2, where each agent chooses with a given probability its next information state over time from a set of possible values obtained from its neighbours. However, this model lacks precision in terms of analysis and determinism, since there is no known practical model to describe the information spreading over each node when there is more than two distinct information in the system, and its spreading rule is purely stochastic. The exact N^n-state Markov chain often used to do that in small systems is too complex to be employed for even a few numbers of nodes, being impracticable at real-world problems. Due to that, precise control of such a spreading and consensus convergence is a complicated task that might be even impossible, concerning generalised networked systems.

So, instead of the original asynchronous Voter Model presented earlier, this work uses the synchronous version of such a model, defined in terms of Equation (2.8) as

$$x_i[k+1] = x_j[k] \quad \text{w.p.} \quad p_{ij} = \frac{a_{ij}}{\delta_i} \qquad i, j = 1, \ldots, n \tag{4.1}$$

where $x_i \in \mathbb{S}$ is the discrete-state information for an agent i and $p_{ij} \in [0, 1]$ is the probability of agent i chooses information from node j. At first sight, it might seem the same model presented in Equation (2.8), but it differs in the sense that every node executes this spreading rule for each time step k, composing a synchronous process.

The reasons to use this synchronous version of the Voter Model instead of the original asynchronous protocol is that, from this spreading rule, it is possible to deduce precise analytical models that can describe the information spreading over any interconnected system in terms of its topology. Such models unleash the power of the topology manipulation to control the information diffusion and consequently, consensus behaviour in generalised networked systems. To eliminate any ambiguity in the terminology, from now on this text refers to this synchronous version of the Voter Model and its dynamical spreading model as Discrete-Space Random Consensus Protocol (DSRCP).

Due to the random nature of x_i, the information dynamics described by Equation (4.1) is uncertain for each experiment[1]. Thus, there is no way to predict precisely, for a performed experiment, the information state of each agent at a given time instant. The next section introduces the approximated DSRCP's dynamical spreading model that is based on epidemics and is used to estimate the probability of occurrence of a particular information $\chi \in \mathbb{S}$ in each agent of the network.

[1]In this text, experiment means a complete execution of the DRSCP algorithm in which all agents reach consensus over some information if it is feasible.

4.1.2. Dynamical Model for Information Spreading

To develop the information spreading model of the decision-making process, concerning the uncertain nature of the information state $x_i[k]$, and the decision rule represented by Equation (4.1), firstly, it is considered the dynamics of the spreading of a single information χ throughout the network, for a given set of initial conditions $\boldsymbol{x}_0 = [x_1[0],\ x_2[0],\ \ldots,\ x_n[0]]^\intercal$. This model is similar to epidemic models that describe the evolution of individual infection probabilities of the agents by a virus χ spreading over the network (GOFFMAN; NEWILL, 1964; MIEGHEM et al., 2009).

In this model, each agent i is associated with a Markov chain with two states, namely, $x_i = \chi$ (infected with information χ) and $x_i = \overline{\chi}$ (not infected with information χ), as depicted in Figure 4.1. It is similar to the epidemic model with concurrent dynamics presented in (LIU et al., 2016; YANG et al., 2018).

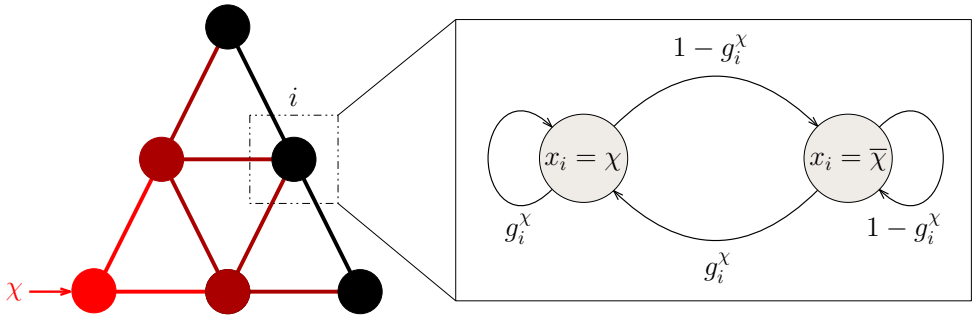


Figure 4.1: DSRCP's individual discrete-time Markov chain for an agent i: x_i is the information state of agent i, χ is the information of interest, and $\overline{\chi}$ is any other information but χ.

For this Markov chain, the individual probability of infection of a agent i by an information χ at time step k is denoted by $z_i^\chi[k] \triangleq \Pr\{x_i[k] = \chi\}$, and the probability of not being infected, is given by $1 - z_i^\chi[k] \triangleq \Pr\{x_i[k] = \overline{\chi}\}$. Considering the Markov chain depicted in Figure 4.1, the evolution of the probability of infection $z_i^\chi[k]$ is given by the following Markov difference equation

$$z_i^\chi[k+1] = (1 - z_i^\chi[k])g_i^\chi[k] + z_i^\chi[k]g_i^\chi[k] \qquad i = 1, \ldots, n \tag{4.2}$$

where $g_i^\chi[k] \in \mathbb{R}$ is the probability of transition from state $x_i[k] = \overline{\chi}$ to state $x_i[k+1] = \chi$ or to remain at state $x_i[k] = \chi$ (i.e. $x_i[k+1] = x_i[k] = \chi$). The transition probability from state $x_i[k] = \chi$ to state $x_i[k+1] = \overline{\chi}$ or to stay at $x_i[k] = \overline{\chi}$ (i.e. $x_i[k+1] = x_i[k] = \overline{\chi}$) is given by $1 - g_i^\chi[k]$.

The transition probability $g_i^\chi[k]$ for a agent i is driven by the weight of the neighbours infected with χ at time k, which can be approximated by the union of the infection probabilities of such neighbours (i.e. the expected amount of infected neighbours at time k, as stated in Mieghem et al. (2009)), yielding

$$g_i^\chi[k] = \mathrm{E}\left\{\beta_i \sum_{j=1}^{n} a_{ij} \mathbb{1}_{\{x_j[k]=\chi\}}\right\} = \beta_i \sum_{j=1}^{n} a_{ij} z_j^\chi[k] \qquad i = 1, \ldots, n \tag{4.3}$$

where $\mathbb{1}_{\{x_j[k]=\chi\}} = 1$ if agent j is infected with information χ at time k, and equal to 0, otherwise; $\mathrm{E}\{\cdot\} : \mathbb{R} \to \mathbb{R}$ is the mathematical expectation of a random variable; and $\beta_i \in \mathbb{R}$ is the *infection rate*.

By taken Equation (4.1) into account, considering $\beta_i = 1/\delta_i$ (i.e. as the inverse of the in-degree of agent i); substituting Equation (4.3) on Equation (4.2) and performing some algebraic simplifications, the infection probability dynamics for information χ in each agent i is given by the following linear difference equation

$$z_i^{\chi}[k+1] = \frac{1}{\delta_i} \sum_{j=1}^{n} a_{ij} z_j^{\chi}[k] \qquad i = 1, \ldots, n. \tag{4.4}$$

If matrix $\boldsymbol{\mathcal{A}}$ is unweighted, this equation ensures all neighbours of a agent i has the same influence over it.

Besides z_i^{χ}, some works such as (MIEGHEM et al., 2009; MIEGHEM, 2011) propose a global network state that could also be used to characterise the spreading of information χ, the so-called *expected fraction of infected nodes*, denoted by $\tilde{y}^{\chi}[k]$ and given by

$$\tilde{y}^{\chi}[k] = \mathrm{E}\{y^{\chi}[k]\} = \mathrm{E}\left\{ \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}_{\{x_i[k]=\chi\}} \right\} = \frac{1}{n} \sum_{i=1}^{n} z_i^{\chi}[k] \tag{4.5}$$

where $y^{\chi}[k] \in [0, 1]$ is a random variable that represents the fraction of infected nodes by information χ at time k, for a given experiment.

The values of $z_i^{\chi}[k]$ and $\tilde{y}^{\chi}[k]$ are estimations for the expected occurrence of information χ and they are fundamentally dependent on the number of performed experiments, according to the law of large numbers (PAPOULIS; PILLAI, 2002). Thus, as many experiments are performed, the lower is the error between their values and the real occurrence.

4.1.3. Existence and Characterisation of Equilibria

By writing Equation (4.4) in matrix form regarding all agents in the network, one have

$$\boldsymbol{z}^{\chi}[k+1] = \boldsymbol{P}\boldsymbol{z}^{\chi}[k] \tag{4.6}$$

where $\boldsymbol{z}^{\chi}[k] = [z_1^{\chi}[k],\ z_2^{\chi}[k],\ \ldots,\ z_n^{\chi}[k]]^{\mathsf{T}}$ is the array of individual spreading probabilities for information χ at time k, $\boldsymbol{P} = \boldsymbol{\Delta}^{-1}\boldsymbol{\mathcal{A}}$ is the system transition matrix (the row-normalised version of $\boldsymbol{\mathcal{A}}$), and $\boldsymbol{\Delta} = \mathrm{diag}(\boldsymbol{\mathcal{A}}\mathbf{1}_n)$ is the valency matrix of the network underlying digraph.

The solution of the discrete-time dynamical linear system represented by Equation (4.6) is

$$\boldsymbol{z}^{\chi}[k] = \boldsymbol{P}^k \boldsymbol{z}_0^{\chi} \tag{4.7}$$

where $\boldsymbol{z}_0^{\chi} = [z_1^{\chi}[0],\ z_2^{\chi}[0],\ \ldots,\ z_n^{\chi}[0]]^{\mathsf{T}}$ is the array of initial individual spreading probabilities for information χ. Note that, the elements of $\boldsymbol{P}$ are the probabilities p_{ij} described in Equation (4.1), i.e. the probability of each agent i chooses the information state of its respective neighbour j.

Equation (4.6) has two trivial equilibrium points, $\boldsymbol{z}_{\infty}^{\chi} = \mathbf{1}_n$ (all agents are infected by χ), and $\boldsymbol{z}_{\infty}^{\chi} = \mathbf{0}_n$ ($\overline{\chi}$ infects all), where $\boldsymbol{z}_{\infty}^{\chi} = \lim_{k\to\infty} \boldsymbol{z}^{\chi}[k]$. Besides these trivial equilibrium points, it is important to verify under which conditions, for a given number of initial infected states, the whole network would be infected by χ, leading to an *endemic state* representing a consensus of all agents regarding the information χ. Theorem 4.1 instantiate necessary conditions to achieve consensus over χ with non-null probability.

Theorem 4.1. *If $\boldsymbol{P}$ is a primitive[2] and row-stochastic matrix, for a given distribution of probabilities $\boldsymbol{z}_0^\chi$, corresponding to an information $\chi \in \mathbb{S}$, then Equation (4.6) has a unique non-trivial equilibrium point for its spreading probabilities, such that $\boldsymbol{z}_\infty^\chi = \boldsymbol{P}\boldsymbol{z}_\infty^\chi$, with $\boldsymbol{z}_\infty^\chi = \gamma^\chi \mathbf{1}_n$, and $\gamma^\chi \in (0, 1)$, corresponding to the probability of the consensus in χ, represented by $\boldsymbol{x}_\infty = \chi \mathbf{1}_n$.*

Proof. The proof of the non-trivial equilibrium point's uniqueness is due to the Perron-Frobenius theorem (GODSIL; ROYLE, 2001) and by the fact that $\boldsymbol{P}$ is a row-stochastic and primitive matrix, which yields the following ascending distribution for its eigenvalues:

$$|\lambda_1| \leq |\lambda_2| \leq \ldots < \lambda_n = 1$$

i.e. its largest eigenvalue has magnitude 1 with algebraic multiplicity one.

That implies the solution of Equation (4.7) when k goes to ∞ and, consequently, the non-trivial equilibrium point of the individual spreading probabilities of χ is fully determined by the convex combination of the leading left and right-eigenvectors of $\boldsymbol{P}$ and $\boldsymbol{z}_0^\chi$, as can be seen by rewriting Equation (4.7) regarding the spectral decomposition of $\boldsymbol{P}$ and taking its limit at infinity:

$$\begin{aligned}
\lim_{k\to\infty} \boldsymbol{z}^\chi[k] &= \boldsymbol{P}^k \boldsymbol{z}_0^\chi \\
&= (\boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^{-1})^k \boldsymbol{z}_0^\chi \\
&= \boldsymbol{Q}\boldsymbol{\Lambda}^k\boldsymbol{Q}^{-1}\boldsymbol{z}_0^\chi \\
&= \left(\boldsymbol{q}_1 \cancel{\lambda_1^k}^{\,0} \bar{\boldsymbol{q}}_1 + \boldsymbol{q}_2 \cancel{\lambda_2^k}^{\,0} \bar{\boldsymbol{q}}_2 + \ldots + \boldsymbol{q}_n \cancel{\lambda_n^k}^{\,1} \bar{\boldsymbol{q}}_n\right) \boldsymbol{z}_0^\chi \\
&= (\boldsymbol{q}_n \bar{\boldsymbol{q}}_n) \boldsymbol{z}_0^\chi
\end{aligned}$$

where $\boldsymbol{Q} = [\boldsymbol{q}_1,\ \boldsymbol{q}_2,\ \ldots,\ \boldsymbol{q}_n]$ is a matrix consisting of the ℓ_2-normalised right-eigenvectors of $\boldsymbol{P}$, $\bar{\boldsymbol{q}}_i \in \mathbb{R}^{1\times n}$ is the i-th row of matrix $\boldsymbol{Q}^{-1}$, $\boldsymbol{\Lambda} = \text{diag}([\lambda_1\ \lambda_2\ \ldots\ \lambda_n])$ is a diagonal matrix with the ordered eigenvalues of $\boldsymbol{P}$ in its diagonal.

As $\lambda_n = 1$ is unique, there is only one possible value for the leading left-eigenvector $\bar{\boldsymbol{q}}_n$, named as $\boldsymbol{v}$. Also, due to the fact $\boldsymbol{P}$ is row-stochastic, $\boldsymbol{q}_n = \mathbf{1}_n/\sqrt{n}$ is its ℓ_2-normalised leading right-eigenvector. That implies the tensor resultant of the operation $\boldsymbol{q}_n\bar{\boldsymbol{q}}_n$ performs a convex combination of $\boldsymbol{v}$ and $\boldsymbol{z}_0^\chi$, yielding

$$\begin{aligned}
\boldsymbol{z}_\infty^\chi &= \frac{\mathbf{1}_n}{\sqrt{n}} \boldsymbol{v}^\intercal \boldsymbol{z}_0^\chi \\
&= \frac{\mathbf{1}_n}{\sqrt{n}} \begin{bmatrix} v_1 & v_2 & \ldots & v_n \end{bmatrix} \begin{bmatrix} z_1^\chi[0] \\ z_2^\chi[0] \\ \vdots \\ z_n^\chi[0] \end{bmatrix} \\
&= \frac{\mathbf{1}_n}{\sqrt{n}} \left(v_1 z_1^\chi[0] + v_2 z_2^\chi[0] + \ldots + v_n z_n^\chi[0]\right) \\
&= \frac{\mathbf{1}_n}{\sqrt{n}} \sum_{i=1}^{n} v_i z_i^\chi[0] = \frac{\mathbf{1}_n}{\|\boldsymbol{v}\|_1} \sum_{i=1}^{n} v_i z_i^\chi[0] = \mathbf{1}_n \gamma^\chi
\end{aligned} \tag{4.8}$$

[2] A matrix is said to be primitive if it is non-negative, irreducible and has only one eigenvalue in its spectral circle, i.e. its largest eigenvalue has algebraic multiplicity one (cf. Chapter 2).

where $\gamma^{\chi} = \boldsymbol{v}^{\mathsf{T}}/\|\boldsymbol{v}\|_1 \boldsymbol{z}_0^{\chi}$ is the convex combination of the ℓ_1-normalised version of $\boldsymbol{v}$ and $\boldsymbol{z}_0^{\chi}$, $\|\cdot\|_1$ is the ℓ_1-norm, and v_i is the i-th element of $\boldsymbol{v}$. Since $\boldsymbol{v}$ is unique, γ^{χ} is also unique regarding $\boldsymbol{z}_0^{\chi}$.

The proof that γ^{χ} is the probability of the consensus over information χ, i.e. $\gamma^{\chi} = \Pr\{\boldsymbol{x}_{\infty} = \chi \mathbf{1}_n\}$, is divided in three parts. Firstly, one shows that the expected fraction of infected nodes at steady-state, $\tilde{y}_{\infty}^{\chi}$, is equal to γ^{χ} by extending Equation (4.5) when k goes to ∞ and by replacing $\boldsymbol{z}_{\infty}^{\chi}$ with the definition of Equation (4.8), yielding

$$\begin{aligned}\tilde{y}_{\infty}^{\chi} &= \mathrm{E}\left\{\frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{\{x_i[\infty]=\chi\}}\right\} \\ &= \frac{1}{n}\sum_{i=1}^{n} z_i^{\chi}[\infty] = \frac{\mathbf{1}_n^{\mathsf{T}}}{n}\boldsymbol{z}_{\infty}^{\chi} \\ &= \frac{\mathbf{1}_n^{\mathsf{T}}\mathbf{1}_n}{n}\gamma^{\chi} = \gamma^{\chi}\end{aligned}$$

where $x_i[\infty] \in \mathbb{S}$ is the steady-state of agent i's information state.

Secondly, by analysing Equation (4.1), the only way of ensuring a steady-state to the node i's information state (i.e. $x_i[k+1] = x_i[k]$, $\forall k \geq \tau$, where $\tau \in \mathbb{N}$ is some positive constant) is by ensuring that all agents have the same information of i (i.e. they have reached consensus) or by assuming $p_{ij} = 0$, $\forall j \in \mathcal{V}$, $i \neq j$, what is only possible if matrix $\boldsymbol{P}$ is not primitive, violating the theorem's statement. That implies the consensus is the only possible outcome for the steady-state of Equation (4.1) with $\boldsymbol{P}$ being primitive.

Thirdly, it is used the fact that the fraction of infected nodes by information χ at steady-state, $y_{\infty}^{\chi} = 1/n \sum_{i=1}^{n} \mathbb{1}_{\{x_i[\infty]=\chi\}}$, is either 1 (when the consensus converges to χ) or 0 (when the consensus converges to $\overline{\chi}$) for a given experiment, i.e. $y_{\infty}^{\chi} = \mathbb{1}_{\{\boldsymbol{x}_{\infty}=\chi\mathbf{1}_n\}}$. Which implies $\tilde{y}_{\infty}^{\chi}$ assumes precisely the ratio between the number of times the consensus happens over χ, and the total number of experiments. That is the probability of consensus in χ, as stated in (MIEGHEM, 2006) for the definition of the expected value of indicator functions, resulting directly in

$$\gamma^{\chi} = \mathrm{E}\{\mathbb{1}_{\{\boldsymbol{x}_{\infty}=\chi\mathbf{1}_n\}}\} = \Pr\{\boldsymbol{x}_{\infty} = \chi\mathbf{1}_n\}$$

which concludes the proof. □

The vector $\boldsymbol{v}$ is known as *eigenvector centrality score* and it is a way to measure every agent's importance in a graph according to their spacial position on there: agents whose neighbours have a high out-degree (i.e. a large number of links starting from it) are more influential than others (BONACICH, 1972). That implies the non-trivial equilibrium point for each information χ's spreading probabilities is defined directly by the relationship of the network centrality of each agent and their initial infection probabilities.

As one can see, the spacial position of an infected agent in the network is determinant to the final value of the infection probabilities of its information. That means the eigenvector centrality score is a measure of each agent's "spreading power" for this consensus algorithm.

4.1.4. Consensus Convergence Criteria

A main consequence of Theorem 4.1 is on the relation between the expected fraction of infected nodes at steady-state by an information χ, $\tilde{y}_{\infty}^{\chi}$, and the consensus probability

of χ. From such relation, one concludes the consensus over the information states is always possible when matrix $\boldsymbol{P}$ is primitive, being driven by the eigenvector centrality score of each node, i.e. the information of nodes that have more "spreading power" are more likely to be prevalent when consensus is reached.

However, the consensus may happen even when $\boldsymbol{P}$ is a non-primitive (or imprimitive) matrix, since its largest eigenvalue keeps its algebraic multiplicity equals to one. For instance, in digraphs with one source vertex (i.e. a node that does not receive information from any other, but that has at least one node receiving its information) the consensus converges to the information of the source node. In general, if a digraph $\mathcal{G}$ has at least a directly spanning tree in its structure, the consensus may happen (being driven by the root of the tree), but there are no guarantees. In all other cases the consensus does not happen for sure.

4.1.5. Convergence Rate of the Infection Probabilities

The convergence rate of the infection probabilities in the DSRCP model described by Equation (4.6) is mainly driven by the second largest eigenvalue of $\boldsymbol{P}$. In fact, as its larger eigenvalue is always equal to 1 (being invariant to k), the last residual of the transition phase until the steady-state be reached is generated by $\lambda_{n-1}(\boldsymbol{P})$. That implies the exactly iteration step τ, where the infection probabilities for information χ converges to its equilibrium is defined in terms of such eigenvalue as

$$\tau = \left\lceil \frac{\ln(\epsilon)}{\ln(\lambda_{n-1}(\boldsymbol{P}))} \right\rceil$$

where $\epsilon \approx 0$ is the value of $\lambda^{\tau}_{n-1}(\boldsymbol{P})$. As $\epsilon > 0$, the equilibrium is not reached at $k = \tau$ since the convergence of the infection probabilities is defined asymptotically. However, one can write those probabilities at this time step concerning an upper bounded residual or error to the endemic state, regarding the initial infection distribution of information χ, yielding

$$\begin{aligned}
z_i^{\chi}[\tau] &= z_1^{\chi}[0]\sum_{j=1}^{n} q_{ij}\lambda_j^{\tau}\bar{q}_{j1} + z_2^{\chi}[0]\sum_{j=1}^{n} q_{ij}\lambda_j^{\tau}\bar{q}_{j2} + \ldots + z_n^{\chi}[0]\sum_{j=1}^{n} q_{ij}\lambda_j^{\tau}\bar{q}_{jn} \\
&\leq z_1^{\chi}[0]\left(\sum_{j=1}^{n-1} q_{ij}\cancelto{\epsilon}{\lambda_j^{\tau}}\bar{q}_{j1} + q_{in}\cancelto{1}{\lambda_n^{\tau}}\bar{q}_{n1}\right) + \ldots + z_n^{\chi}[0]\left(\sum_{j=1}^{n-1} q_{ij}\cancelto{\epsilon}{\lambda_j^{\tau}}\bar{q}_{jn} + q_{in}\cancelto{1}{\lambda_n^{\tau}}\bar{q}_{nn}\right) \\
&\leq q_{in}\sum_{j=1}^{n}\bar{q}_{nj}z_j^{\chi}[0] + \epsilon\sum_{l=1}^{n}\sum_{j=1}^{n-1} q_{ij}\bar{q}_{jl}z_l^{\chi}[0] \\
&\leq q_{in}\bar{\boldsymbol{q}}_n\boldsymbol{z}^{\chi}[0] + \epsilon\sum_{j=1}^{n-1} q_{ij}\bar{\boldsymbol{q}}_j\boldsymbol{z}^{\chi}[0] \\
&< z_i^{\chi}[\infty] \pm \epsilon\mathbf{1}_n^{\mathsf{T}}\boldsymbol{z}^{\chi}[0].
\end{aligned}$$

That implies the residual is upper bounded by the number of nodes infected with information χ at time step $k = 0$ multiplied by ϵ, i.e. $\epsilon\mathbf{1}_n^{\mathsf{T}}\boldsymbol{z}^{\chi}[0]$. Moreover, this residual still over estimated, mainly because in arbitrary graphs, there is a gap between the second largest eigenvalue of $\boldsymbol{P}$ and the others, lower in absolute magnitude, which imply their residuals tend to be much lower than ϵ at time step τ. Thus, one can interpret τ as the discrete time step where the difference between $\boldsymbol{z}^{\chi}[\tau]$ and the steady-state $\boldsymbol{z}^{\chi}_{\infty}$ is assuredly

lower than $\epsilon \mathbf{1}_n^\intercal \boldsymbol{z}^\chi[0]$, i.e.

$$\tau \in \mathbb{N} : |z_i^\chi[\infty] - z_i^\chi[\tau]| < \epsilon \mathbf{1}_n^\intercal \boldsymbol{z}^\chi[0],\ \forall i \in \mathcal{V}.$$

Note that, as τ depends only on $\lambda_{n-1}(\boldsymbol{P})$, it is invariant for each information χ contained in $\boldsymbol{x}_0$. But the magnitude of the residual to the steady-state of the infection probabilities at τ varies concerning each information infection probability at the beginning.

4.1.6. Multi-Epidemic Spreading Model

This section extends the results on the DSRCP epidemic deterministic model over a single piece of information in $\boldsymbol{x}_0$ to a multi-epidemic model, that describes the spreading probability of all the unique information contained in $\boldsymbol{x}_0$, simultaneously.

Equation (4.6) is rewritten, concerning all the distinct information contained in $\boldsymbol{x}_0$, yielding

$$\boldsymbol{Z}[k+1] = \boldsymbol{P}\boldsymbol{Z}[k] \tag{4.9}$$

where $\boldsymbol{Z}[k] = [\boldsymbol{z}^{\chi_1}[k],\ \boldsymbol{z}^{\chi_2}[k],\ \ldots,\ \boldsymbol{z}^{\chi_N}[k]] \in \mathbb{R}^{n \times N}$ is the multi-dimensional array of information spreading probabilities for each distinct information $\chi_i \in \boldsymbol{x}_0$, where $i = 1, \ldots, N$, and $N \leq n$ is the number of unique discrete information in $\boldsymbol{x}_0$. Variable $\boldsymbol{Z}[k]$ is the distribution of spreading probabilities for all unique information contained in $\boldsymbol{x}_0$. That implies, it is a row-stochastic matrix, i.e. $\boldsymbol{Z}[k]\mathbf{1}_N = \mathbf{1}_n,\ \forall k \in \mathbb{N}$.

In practice, the multi-epidemic dynamics of the DSRCP is just a generalisation over all the information set $\boldsymbol{x}_0$ of the single-epidemic model proposed on last sections. It still has the same convergence and spectral solution depending on the topology, as $\boldsymbol{P}$ is also its transition matrix. The advantage of this model over the single epidemic one is that it can handle all the information's spreading probabilities at once.

Example 17 presents the DSRCP analysis over a multi-agent system whose underlying network topology is primitive. It shows how the epidemic model correctly predicts the individual infection ratio for all information in $\boldsymbol{x}_0$, and that the steady-state of the infection probabilities indicate the consensus probabilities for each information.

Example 4.1.1. Be a multi-agent system trying to solve a discrete-space consensus problem by executing Equation (4.1) and whose underlying network topology is defined by the direct graph shown in Figure 4.2. The self-loops indicate that each agent considers its information during their state update. The edges' weights indicate the probability of the agent at the edge's head chooses the information of the agent at the edge's tail. The adjacency and transition matrices that describe such a topology are defined as

$$\boldsymbol{\mathcal{A}} = \begin{bmatrix} 0.63 & 0.62 & 1 \\ 0.87 & 1 & 0 \\ 0 & 0.98 & 1 \end{bmatrix} \qquad \boldsymbol{P} = \begin{bmatrix} 0.28 & 0.28 & 0.44 \\ 0.46 & 0.54 & 0 \\ 0 & 0.49 & 0.51 \end{bmatrix}.$$

The eigenvalues of $\boldsymbol{P}$ are $\lambda(\boldsymbol{P}) = [0.16 - 0.3i,\ 0.16 + 0.3i,\ 1]$ and the eigenvector centrality score is $\boldsymbol{v}/\|\boldsymbol{v}\|_1 = [0.29,\ 0.45,\ 0.26]^\intercal$, indicating the less and the most influential agents are 3 and 2, respectively.

Assume the agents have their discrete information defined as $\boldsymbol{x}_0 = [\mathrm{A},\ \mathrm{B},\ \mathrm{C}]$ at the beginning of the DSRCP execution. So, the distribution of initial infection probabilities

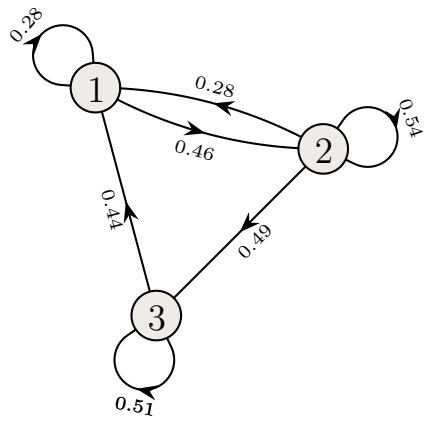


Figure 4.2: Network with $n = 3$ agents. The self-loops indicate each agent considers its own information during their state update. The edges' weights indicate the probability of the agent at the edge's head chooses the information of the agent at the edge's tail.

for all information in $\boldsymbol{x}_0$ is

$$\boldsymbol{Z}_0 = \begin{array}{c} \begin{matrix} \text{A} & \text{B} & \text{C} \end{matrix} \\ \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{array} \begin{matrix} 1 \\ 2 \\ 3 \end{matrix}$$

as every agent has a distinct information. By iteratively execute Equation (4.9) given $\boldsymbol{Z}_0$ and $\boldsymbol{P}$, one get the time evolution of all individual infection probabilities for each information in $\boldsymbol{x}_0$, as can be seeing in Figure 4.3. All infection probabilities reach steady-state, what is expected due to the characteristics of the network topology.

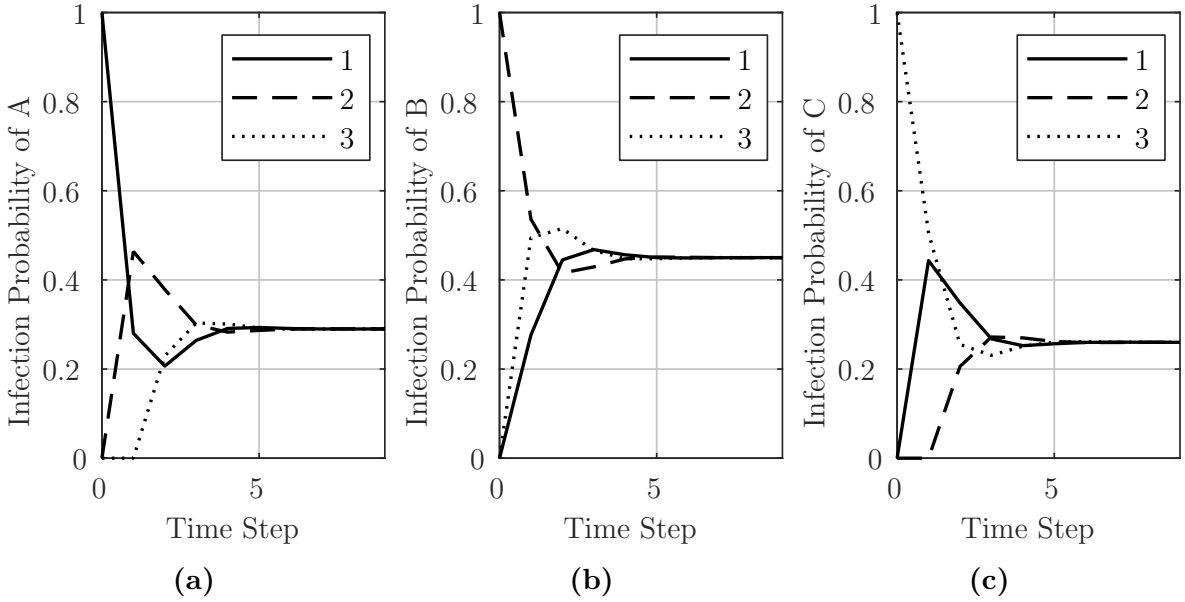


Figure 4.3: Individual infection probabilities for each information at $\boldsymbol{x}_0$: (a) shows the infection probabilities for information A, (b) shows the infection probabilities for information B, (c) shows the infection probabilities for information C. The continuous lines are agent 1's infection probabilities, dashed lines are agent 2's infection probabilities, and dotted lines are agent 3's infection probabilities.

To evaluate if those infection probabilities are precisely enough, one could expose all the $3^3 = 27$ network states derived from every possible combination of information distribution given $\boldsymbol{x}_0$ and calculate the transition probabilities among those states. Although such a strategy would work in this case, it is impracticable for large networks due to combinatorial explosion. Instead, it is performed a simulation for random events following the structure of Algorithm 1. Its inputs are the transition matrix $\boldsymbol{P}$ given from the network topology and the initial information state $\boldsymbol{x}_0$. The algorithm outputs are matrix $\boldsymbol{C} = [c_{iq}] \in \mathbb{N}^{n \times N}$ that contains the occurrence of each information $\chi_q \in \boldsymbol{x}_0$ in each agent $i \in \mathcal{V}$ for all time step $k \in \{0, \ldots, \overline{k}\}$, and array $\boldsymbol{o} = [o_q] \in \mathbb{N}^N$ that contains the occurrence of consensus for each information $\chi_q \in \boldsymbol{x}_0$ for all time step $k \in \{0, \ldots, \overline{k}\}$, where $\overline{k} \in \mathbb{N}$ is the maximum number of DSRCP's iterations. Matrix $\boldsymbol{S} \in \{0, 1\}^{n \times n}$ is an upper triangular matrix with order n, and $\boldsymbol{\mathfrak{P}} = [\mathfrak{p}_{ij}] \in \mathbb{R}^{n \times n}$ is a matrix corresponding the cumulative sum of all columns of matrix $\boldsymbol{P}$. The number of independent experiments is defined by $\bar{l} \in \mathbb{N}$. Function rand$(\cdot)$ generates a uniformly distributed pseudorandom number in the given interval.

Algorithm 1: Simulation of random events for the DSRCP

Input: $\boldsymbol{P}$, $\boldsymbol{x}_0$
Output: $\boldsymbol{C}$, $\boldsymbol{o}$

$\boldsymbol{C}[k] \leftarrow \mathbf{0}_{n \times N},\ \forall k \in \{0, \ldots, \overline{k}\}$ // counter for information occurrence
$\boldsymbol{o}[k] \leftarrow \mathbf{0}_N,\ \forall k \in \{0, \ldots, \overline{k}\}$ // counter for consensus occurrence
$\boldsymbol{x}[0] \leftarrow \boldsymbol{x}_0$ // set the initial information state
$\boldsymbol{\mathfrak{P}} \leftarrow \boldsymbol{PS}$ // cummulative sum of $\boldsymbol{P}$
for $l \leftarrow 1$ **to** $\bar{l}$ **do** // performs $\bar{l}$ independent experiments
    $k \leftarrow 0$ // reset the time step
    while $(\boldsymbol{x}[k] \neq \chi_q \mathbf{1}_n,\ q \in \{1, \ldots, N\})$ **and** $k \leq \overline{k}$ **do**
        for $i \leftarrow 1$ **to** n **do** // simulate the agents' interaction
            $r \leftarrow \text{rand}([0, 1])$ // gets a uniformly distributed random number
            $j \leftarrow \min\{j \in \mathcal{V},\ \text{s.t. } \mathfrak{p}_{ij} \geq r\}$ // gets j with probability p_{ij}
            $x_i[k+1] \leftarrow x_j[k]$ // updates the information state of i
            if $x_i[k] = \chi_q,\ q \in \{1, \ldots, N\}$ **then**
                $c_{iq}[k] \leftarrow c_{iq}[k] + 1$ // occurrence of χ_q in i at time k
        if $\boldsymbol{x}[k] = \chi_q \mathbf{1}_n,\ q \in \{1, \ldots, N\}$ **then**
            $o_q[k] \leftarrow o_q[k] + 1$ // occurrence of consensus at time k
        $k \leftarrow k + 1$ // increase the time step
return $\boldsymbol{C}$, $\boldsymbol{o}$

In this case, the execution of Algorithm 1 with parameters $\bar{l} = 10000$ and $\overline{k} = 10$, results in a simulation set where each agent in the network runs 10 iterations of Equation (4.1), performing an experiment. Such a procedure is repeated independently for 10000 times, counting the number of times every information of $\boldsymbol{x}_0$ reaches each node at each time step k, which is obtained from matrix $\boldsymbol{C}$. The trajectories of $c_{iq}[k]/10000,\ \forall i \in \mathcal{V},\ \forall q \in \{1, \ldots, 3\},\ \forall k \in \{0, \ldots, 10\}$ should be similar to the corresponding infection probabilities described in Figure 4.3 to ensure that the proposed dynamical model correctly describes the information diffusion of the DSRCP. Note that, in this case, $\boldsymbol{x}_0$ is constant for all the experiments since there are no uncertainties over the initial information of each

node.

Figure 4.4 depicts the trajectories obtained from $c_{iq}[k]/10000$, $\forall i \in \mathcal{V}$, $\forall q \in \{1, \ldots, 3\}$, $\forall k \in \{0, \ldots, 10\}$. As can be seeing, they are very similar to the trajectories of the individual infection probabilities depicted in Figure 4.3. Figure 4.5 shows the error between those values for each information in $\boldsymbol{x}_0$. Note that, the maximum error between them is always lower than 0.01, indicating the accuracy of the proposed spreading model.

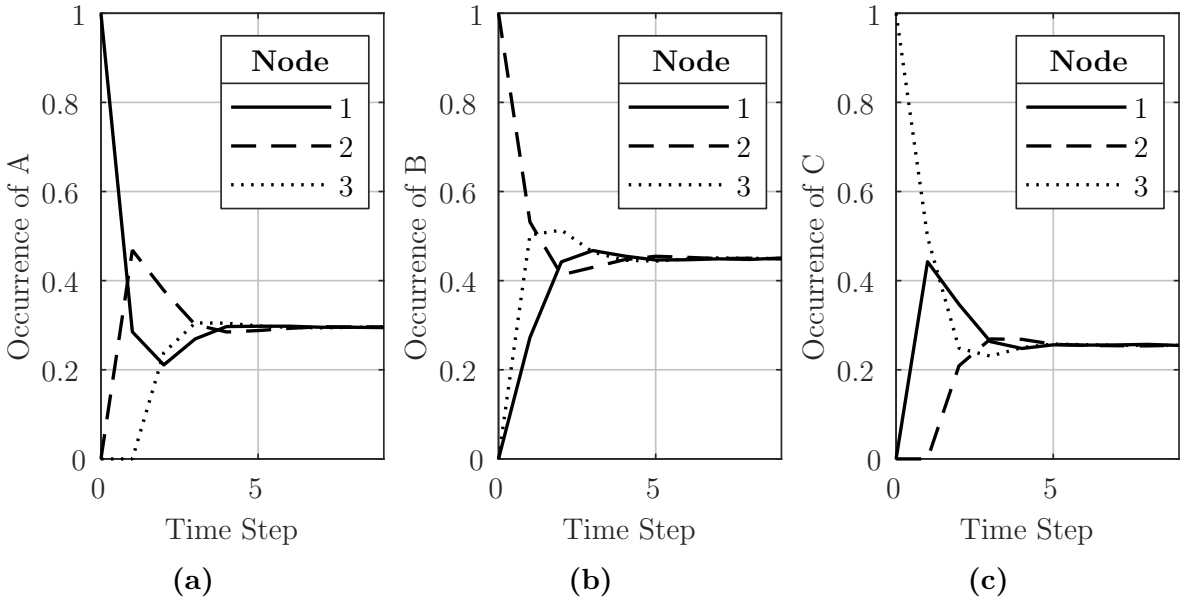


Figure 4.4: Individual information occurrence over time: (a) shows the occurrence for information A, (b) shows the occurrence for information B, (c) shows the occurrence for information C. The continuous lines are the trajectories for the information occurrence in agent 1, dashed lines are the trajectories for the information occurrence in agent 2, and dotted lines are the trajectories for the information occurrence in agent 3.

By extending Equation (4.8) to the multi-epidemic case, one can get the steady-state of the infection probabilities for all information in $\boldsymbol{x}_0$ directly from $\boldsymbol{Z}_0$ and $\boldsymbol{v}$, yielding

$$\begin{aligned}\boldsymbol{Z}_\infty &= \mathbf{1}_n \frac{\boldsymbol{v}^\intercal}{\|\boldsymbol{v}\|_1} \boldsymbol{Z}_0 \\ &= \begin{bmatrix}1\\1\\1\end{bmatrix} \begin{bmatrix}0.29 & 0.45 & 0.26\end{bmatrix} \begin{bmatrix}1 & 0 & 0\\0 & 1 & 0\\0 & 0 & 1\end{bmatrix} = \begin{bmatrix}0.29 & 0.45 & 0.26\\0.29 & 0.45 & 0.26\\0.29 & 0.45 & 0.26\end{bmatrix}\end{aligned}$$

which implies the consensus happens with probability 0.29 for information A, 0.45 for information B, and 0.26 for information C. Note as these values match with the steady-state values presented in Figure 4.3.

To evaluate those values, three series of independent experiments are performed by executing Algorithm 1 with $\bar{l} \in \{100, 1000, 10000\}$ and $\overline{k} = \infty$. Figure 4.6a depicts the consensus occurrence, $\sum_{k=0}^{\overline{k}} o_q[k]/\bar{l}$, $\forall q \in \{1, \ldots, 3\}$, for each experiment versus the steady-state of the infection probabilities (represented by the plus symbols), and Figure 4.6b present the error between the consensus occurrence for each experiment and the steady-state of the infection probabilities. It is evident the consensus occurrence matches pretty well the consensus probabilities estimated earlier from the eigenvector centrality.

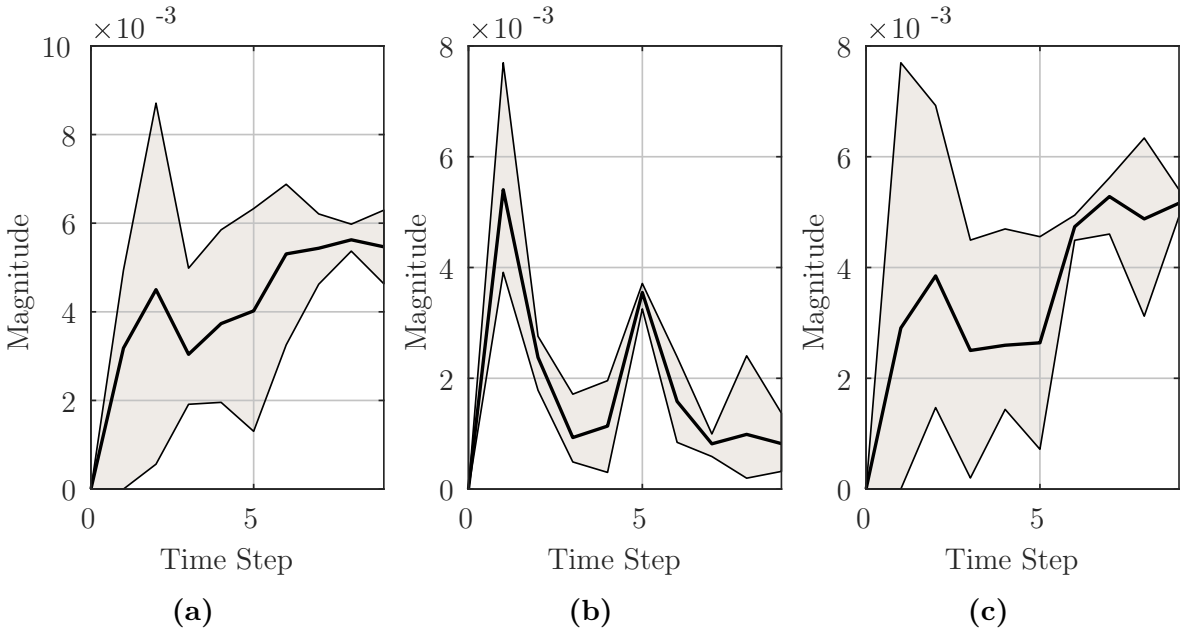


Figure 4.5: Error between the trajectories of the infection probabilities and the information occurrence: (a) shows the error for information A, (b) shows the error for information B, (c) shows the error for information C. The grey areas indicate the variance between the maximum and minimum errors among all nodes, while the black bold line is the average error.

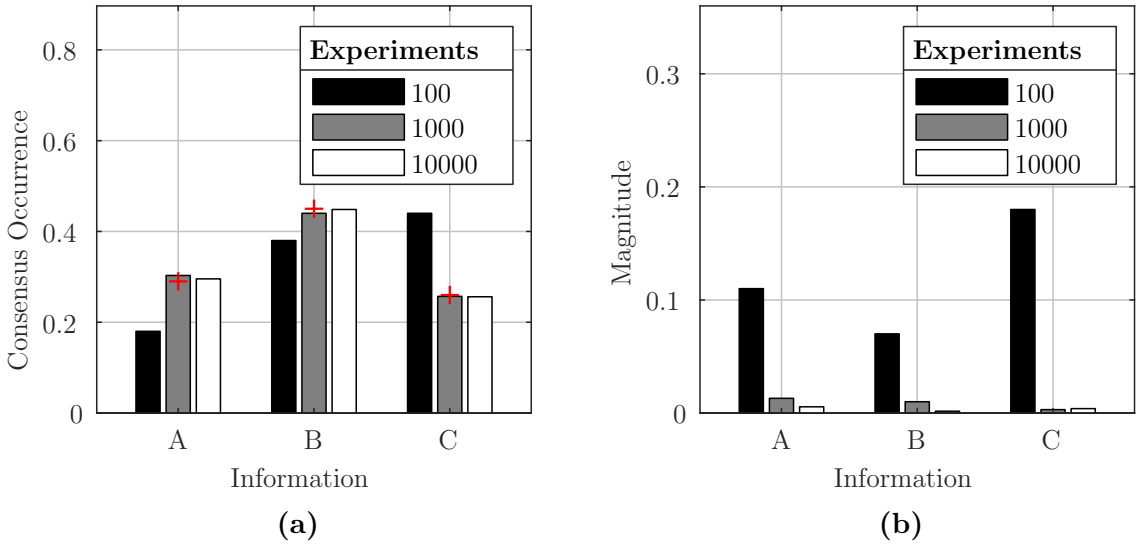


Figure 4.6: Consensus occurrence for each information: (a) shows the histogram of consensus occurrence for each information, and (b) shows the error between the consensus occurrence in each experiment and the steady-state of the infection probabilities. The plus symbols indicate the steady-state of the infection probabilities presented in Figure 4.3.

4.2. Network Weight Design for Manipulation of Consensus

By looking at Equation (4.9), one can see that the spreading probabilities are mainly defined by the underlying network topology, captured by matrix $\boldsymbol{P}$. So, according to the topology, the same initial infection probabilities $\boldsymbol{Z}_0$ may lead to different steady-states $\boldsymbol{Z}_\infty$,

and such values influence the convergence of the consensus directly. Even by changing only the weights of the links in a network and, by consequence, the influence of each agent in its neighbourhood, it is possible to manipulate matrix $\boldsymbol{P}$ and consequently reach almost any $\boldsymbol{Z}_\infty$. Hence, the main idea behind the network weight design for manipulation of discrete-space consensus is on reweighting the adjacency matrix aiming to change the social influence of each agent, concerning the desired consensus distribution for all unique information in the system.

This section presents an offline network reweighting approach to drive the steady-state of the DSRCP, given a set of initial distributions for all unique information contained in $\boldsymbol{x}_0$. The epidemic-based spreading model introduced in the last section is used to calculate the information spreading probability of the DSRCP over a multi-agent system, and semidefinite programming (SDP) is used to design the proper weights of an adjacency matrix that allows Equation (4.9) to evolve to a desired endemic state, as stated in Proposition 4.2.1. That is equivalent to drive the consensus convergence to the desired information state with the desired probability defined by the steady-state of the infection probabilities, as shown in Theorem 4.1.

Proposition 4.2.1. *Be a set of countable information* $\boldsymbol{x}_0 = [x_1[0] = \chi_1,\ x_2[0] = \chi_2,\ \ldots, x_n[0] = \chi_N]^\mathsf{T}$ *the initial information state of a network with n nodes under the DSRCP algorithm described by Equation* (4.1)*. There is an optimal adjacency matrix* $\boldsymbol{\mathcal{A}}^\star$ *obtained from a given topology described by the unweighted matrix* $\boldsymbol{\mathcal{A}}_0$ *that can drive the nodes' information state under the DSRCP to a desired stationary distribution* $\boldsymbol{x}_r = [x_1 = \chi_i,\ x_2 = \chi_i,\ \ldots,\ x_n = \chi_i]^\mathsf{T},\ i \in [1, N]$ *with probability* $z_r^{\chi_i} = \Pr\{\boldsymbol{x}_\infty = \boldsymbol{x}_r | \boldsymbol{x}_0\}$.

In other words, there is an optimal adjacency matrix $\boldsymbol{\mathcal{A}}^\star$ defined in such a way that

$$\boldsymbol{Z}_r = \lim_{k\to\infty} \left(\operatorname{diag}(\boldsymbol{\mathcal{A}}^\star \mathbf{1}_n)^{-1}\boldsymbol{\mathcal{A}}^\star\right)^k \boldsymbol{Z}_0$$

where $\boldsymbol{Z}_r = \mathbf{1}_n[z_r^{\chi_1},\ z_r^{\chi_2},\ \ldots,\ z_r^{\chi_N}]$ is the desired distribution of spreading probabilities for all unique information contained in $\boldsymbol{x}_0$. However, matrix $\boldsymbol{\mathcal{A}}^\star$ is restricted to the structure of a given unweighted adjacency matrix $\boldsymbol{\mathcal{A}}_0$ that describes the links of the existing network topology. That implies matrix $\boldsymbol{\mathcal{A}}^\star$ minimises asymptotically the following non-linear function for large values of k

$$\Psi(\boldsymbol{\mathcal{A}}, k) = \|\boldsymbol{Z}_r - \left(\operatorname{diag}((\boldsymbol{\mathcal{A}} \circ \boldsymbol{\mathcal{A}}_0)\mathbf{1}_n)^{-1}(\boldsymbol{\mathcal{A}} \circ \boldsymbol{\mathcal{A}}_0)\right)^k \boldsymbol{Z}_0\|_2$$

in a way that $\lim_{k\to\infty} \Psi(\boldsymbol{\mathcal{A}}^\star, k) = 0$. Operator $\circ$ is the matrix entrywise product, also known as *Hadamard product*. This problem is highly non-linear, non-convex, and cannot be solved by usual optimisation techniques.

The subsequent section presents the main contribution of this chapter: an approach that solves this problem for any given primitive graph described by $\boldsymbol{\mathcal{A}}_0$, any initial infection distribution $\boldsymbol{Z}_0$, and any valid infection distribution $\boldsymbol{Z}_r$.

4.2.1. Optimal Weight Design with Spectral Optimisation

The basic idea behind this approach is on the fact that the non-trivial equilibrium points of the DSRCP are given precisely by the spectrum of $\boldsymbol{P}$, as described in the last section. Instead of searching for the adjacency matrix $\boldsymbol{\mathcal{A}}^\star$ directly, one can look for the corresponding transition matrix $\boldsymbol{P}^\star = \operatorname{diag}(\boldsymbol{\mathcal{A}}^\star \mathbf{1}_n)^{-1}\boldsymbol{\mathcal{A}}^\star$ that drives asymptotically the

infection probabilities from $\boldsymbol{Z}_0$ to $\boldsymbol{Z}_r$, i.e. $\boldsymbol{Z}_r = \lim_{k\to\infty}(\boldsymbol{P}^\star)^k\boldsymbol{Z}_0$. Matrix $\boldsymbol{P}^\star$ is the optimal solution of a semidefinite optimisation program.

By knowing a priori a matrix $\boldsymbol{T} = \boldsymbol{u}\boldsymbol{v}^\mathsf{T}$ resultant from the outer product of vectors $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^n$, that gives a desired infection distribution $\boldsymbol{Z}_r$ directly, i.e. $\boldsymbol{Z}_r = \boldsymbol{T}\boldsymbol{Z}_0$, one can build a transition matrix $\boldsymbol{P}^\star$ such that its leading left and right-eigenvectors are $\boldsymbol{v}^\mathsf{T}$ and $\boldsymbol{u}$, respectively. Such a matrix drives asymptotically the infection probabilities from $\boldsymbol{Z}_0$ to $\boldsymbol{Z}_r$, what is summarized in Theorem 4.2.

Theorem 4.2. *Be a matrix $\boldsymbol{T} \in \mathbb{R}^{n\times n}$ that maps the initial infection distribution $\boldsymbol{Z}_0 = [\boldsymbol{z}^{\chi_1}[0],\ \boldsymbol{z}^{\chi_2}[0],\ \ldots,\ \boldsymbol{z}^{\chi_N}[0]]$ into the desired infection distribution $\boldsymbol{Z}_r = \mathbf{1}_n[z_r^{\chi_1},\ z_r^{\chi_2},\ \ldots,\ z_r^{\chi_N}]$ for N unique information in $\boldsymbol{x}_0$, i.e. $\boldsymbol{Z}_r = \boldsymbol{T}\boldsymbol{Z}_0$. Assume that $\boldsymbol{T}$ can be written as an outer product of two vectors: $\boldsymbol{u} \in \mathbb{R}^n$ and $\boldsymbol{v} \in \mathbb{R}^n$, i.e. $\boldsymbol{T} = \boldsymbol{u}\boldsymbol{v}^\mathsf{T}$. If transition matrix $\boldsymbol{P}$ in Equation (4.9) is primitive, with $\boldsymbol{v}^\mathsf{T}$ and $\boldsymbol{u}$ as left and right-eigenvectors related to its largest eigenvalue, respectively, then one can affirm that the dynamics of the infection probabilities of the DSRCP, given an initial infection distribution $\boldsymbol{Z}_0$, assuredly converge to $\boldsymbol{Z}_r$ as the time goes to infinity, i.e. $\lim_{k\to\infty}\boldsymbol{Z}[k] = \boldsymbol{P}^k\boldsymbol{Z}_0 = \boldsymbol{Z}_r$ which implies that $\lim_{k\to\infty}\boldsymbol{P}^k = \boldsymbol{T}$.*

Proof. Starting from the results of Theorem 4.1, if matrix $\boldsymbol{P}$ is primitive and row-stochastic, then its power converges to a tensor resulting from the outer product of its leading right and left-eigenvectors, i.e. $\lim_{k\to\infty}\boldsymbol{P}^k = \boldsymbol{q}_n\bar{\boldsymbol{q}}_n$. By assuming $\boldsymbol{v}^\mathsf{T}$ and $\boldsymbol{u}$ as leading left and right-eigenvectors of $\boldsymbol{P}$, then it is directly that $\lim_{k\to\infty}\boldsymbol{P}^k = \boldsymbol{u}\boldsymbol{v}^\mathsf{T} = \boldsymbol{T}$, yielding

$$\lim_{k\to\infty}\boldsymbol{Z}[k] = \boldsymbol{P}^k\boldsymbol{Z}_0 = \boldsymbol{T}\boldsymbol{Z}_0 = \boldsymbol{Z}_r$$

which concludes the proof.

□

Matrix $\boldsymbol{T}$ can be written in terms of $\boldsymbol{Z}_0$ and $\boldsymbol{Z}_r$ by extending Equation (4.8) to the multi-epidemic case and doing $\boldsymbol{z}_\infty^\chi = \boldsymbol{z}_r^\chi$, yielding

$$\boldsymbol{T} = \boldsymbol{Z}_r\mathrm{diag}(\boldsymbol{Z}_0^\mathsf{T}\mathbf{1}_n)^{-1}\boldsymbol{Z}_0^\mathsf{T} = \boldsymbol{u}\boldsymbol{v}^\mathsf{T}, \tag{4.10}$$

where $\boldsymbol{u} = \mathbf{1}_n/\sqrt{n}$ is a normalised unit vector, and $\boldsymbol{v}^\mathsf{T} = \boldsymbol{u}^\mathsf{T}\boldsymbol{T}$ is the vector that results in the matrix $\boldsymbol{T}$ when under the outer product with $\boldsymbol{u}$.

Equation (4.10) is not well-defined in case of uncertainties about the initial infection probabilities for some information (i.e. when $\exists\chi \in \boldsymbol{x}_0$, such that for some $i \in \mathcal{V}$, $0 < z_i^\chi[0] < 1$). In these situations, one must use an optimisation program to find the corresponding left-eigenvector $\boldsymbol{v}$, as described ahead.

Given an existing topology described by an adjacency matrix $\boldsymbol{\mathcal{A}}_0$, one can find the values of $\boldsymbol{P}$ that drives the system from the initial distribution of infection probabilities $\boldsymbol{Z}_0$ to the desired $\boldsymbol{Z}_r$ by solving two optimisation problems: i) find vector $\boldsymbol{v}$, such that matrix $\boldsymbol{T}$ is the result of the outer product $\boldsymbol{u}\boldsymbol{v}^\mathsf{T}$; ii) find a transition matrix $\boldsymbol{P}$ restricted to the existing links of $\boldsymbol{\mathcal{A}}_0$ such that $\boldsymbol{u}$ and $\boldsymbol{v}$ are, respectively, its right and left-eigenvectors relative to its largest eigenvalue.

The first optimisation problem is solved by a semidefinite program built around

matrix $\boldsymbol{T}$ and vector $\boldsymbol{v}$, yielding

$$\begin{aligned}
\Upsilon_v(\boldsymbol{Z}_0, \boldsymbol{Z}_r) : \min_{\substack{s\in\mathbb{R},\\ \boldsymbol{\alpha}\in\mathbb{R}^n}} \quad & s && \text{(4.11a)}\\
\text{s.t.} \quad & \boldsymbol{T} = \boldsymbol{Z}_r \text{diag}(\boldsymbol{Z}_0^\mathsf{T}\mathbf{1}_n)^{-1}\boldsymbol{Z}_0^\mathsf{T}\text{diag}(\boldsymbol{\alpha}), && \text{(4.11b)}\\
& \boldsymbol{U} = \boldsymbol{Z}_r - \boldsymbol{T}\boldsymbol{Z}_0, && \text{(4.11c)}\\
& \begin{bmatrix} s\boldsymbol{I}_n & \boldsymbol{U} \\ \boldsymbol{U}^\mathsf{T} & s\boldsymbol{I}_N \end{bmatrix} \succeq 0, && \text{(4.11d)}\\
& \boldsymbol{u} = \frac{1}{\sqrt{n}}\mathbf{1}_n, && \text{(4.11e)}\\
& \boldsymbol{v} = \boldsymbol{T}^\mathsf{T}\boldsymbol{u}, && \text{(4.11f)}\\
& \boldsymbol{v}^\mathsf{T}\boldsymbol{u} = 1 && \text{(4.11g)}
\end{aligned}$$

where $s \in \mathbb{R}$ is a point from Euclidean norm's epigraph, $\boldsymbol{u} \in \mathbb{R}^n$ is a normalised unit vector, $\boldsymbol{v} \in \mathbb{R}^n$ is the vector which the outer product with $\boldsymbol{u}$ results in the matrix $\boldsymbol{T}$ (i.e. $\boldsymbol{T} = \boldsymbol{u}\boldsymbol{v}^\mathsf{T}$), and $\boldsymbol{\alpha} \in \mathbb{R}^n$ is a slack vector to compensates the uncertainty over the initial infection probabilities. Note that, when there are no uncertainties in the distribution of the initial infection probabilities, $\boldsymbol{\alpha} = \mathbf{1}_n$ and $\boldsymbol{U} = \mathbf{0}_{n\times N}$, what implies vector $\boldsymbol{v}$ is obtained directly from Equation (4.10), using $\boldsymbol{T}$ and $\boldsymbol{u}$ without solving the optimisation problem.

Constraint (4.11d) is a linear matrix inequality (LMI) that is equivalent to the second-order cone constraint $\|\boldsymbol{Z}_r - \boldsymbol{T}\boldsymbol{Z}_0\|_2 \leq s$, used to compensate the uncertainties on the initial distribution of infection probabilities for each node through vector $\boldsymbol{\alpha}$. The original constraint is non-linear and non-differentiable at its minimum value, though convex. On the other hand, in the LMI version, the problem resumes itself in minimizing the linear function $\boldsymbol{U}$ over the intersection of an affine set and the cone of positive semidefinite matrices (LOBO et al., 1998).

Constraint (4.11f) ensures $\boldsymbol{T}$ as a tensor resulting from the outer product of $\boldsymbol{u}\boldsymbol{v}^\mathsf{T}$. It comes by the fact that $\boldsymbol{u}^\mathsf{T}\boldsymbol{u} = 1$, and therefore $\boldsymbol{u}^\mathsf{T}\boldsymbol{T} = \boldsymbol{u}^\mathsf{T}\boldsymbol{u}\boldsymbol{v}^\mathsf{T} = \boldsymbol{v}^\mathsf{T}$, which implies in $\boldsymbol{v} = \boldsymbol{T}^\mathsf{T}\boldsymbol{u}$.

Constraint (4.11g) is necessary because vector $\boldsymbol{v}$ found on Program (4.11) should be compatible with the left-eigenvector $\bar{\boldsymbol{q}}_n$ that is taken from the n-th row of the inverse matrix of right-eigenvectors $\boldsymbol{Q}$ of the transition matrix $\boldsymbol{P}$ (as shown in Section 4.1.3) and therefore $\bar{\boldsymbol{q}}_n\boldsymbol{q}_n = 1$ since $\boldsymbol{Q}^{-1}\boldsymbol{Q} = \boldsymbol{I}_n$, implying also that $\bar{\boldsymbol{q}}_i\boldsymbol{q}_i = 1,\ \forall i \in \mathcal{V}$.

Once vector $\boldsymbol{v}$ is found, it is used as a constant in the next optimisation problem, to find the best transition matrix $\boldsymbol{P}$ that drives Equation (4.9) from its initial state to the distribution of desired endemic states, i.e. a matrix $\boldsymbol{P}^\star$ such that $\lim_{k\to\infty}(\boldsymbol{P}^\star)^k = \boldsymbol{u}\boldsymbol{v}^\mathsf{T}$. The transition matrix $\boldsymbol{P}^\star$ has its structure restricted to the links of a given adjacency matrix $\boldsymbol{\mathcal{A}}_0$, so it cannot have any link the given topology does not have, though it can remove existing links by setting zero to their weight.

The second optimisation problem is also solved by a semidefinite program but now

built around the transition matrix $\boldsymbol{P}$, resulting in

$$\begin{aligned}
\Upsilon_P(\boldsymbol{v}, \boldsymbol{\mathcal{A}}_0): \min_{\substack{\boldsymbol{P}\in\mathbb{R}^{n\times n},\\ \alpha,s,\nu\in\mathbb{R}}} \quad & s && \text{(4.12a)}\\
\text{s.t.} \quad & \boldsymbol{P} = \boldsymbol{P} \circ \boldsymbol{\mathcal{A}}_0, && \text{(4.12b)}\\
& \boldsymbol{\mathcal{L}} = \boldsymbol{I}_n - \boldsymbol{P}, && \text{(4.12c)}\\
& \hat{\boldsymbol{\mathcal{L}}} = \frac{\boldsymbol{\mathcal{L}} + \boldsymbol{\mathcal{L}}^\mathsf{T}}{2}, && \text{(4.12d)}\\
& \boldsymbol{U} = \boldsymbol{P}^\mathsf{T}\boldsymbol{v} - \boldsymbol{v} + \boldsymbol{P}\boldsymbol{u} - \boldsymbol{u}, && \text{(4.12e)}\\
& \boldsymbol{P}\mathbf{1}_n = \mathbf{1}_n, && \text{(4.12f)}\\
& \begin{bmatrix} s\boldsymbol{I}_n & \boldsymbol{U} \\ \boldsymbol{U}^\mathsf{T} & s \end{bmatrix} \succeq 0, && \text{(4.12g)}\\
& \alpha\mathbf{1}_n\mathbf{1}_n^\mathsf{T} - \nu\boldsymbol{I}_n + \hat{\boldsymbol{\mathcal{L}}} \succeq 0, && \text{(4.12h)}\\
& 0 \leq p_{ij} \leq 1, \qquad i,j = 1,\ldots,n, && \text{(4.12i)}\\
& \nu > 0 && \text{(4.12j)}
\end{aligned}$$

where $\boldsymbol{\mathcal{L}} \in \mathbb{R}^{n\times n}$ is the Laplacian of the transition matrix, $\boldsymbol{\mathcal{A}}_0 \in \mathbb{R}^{n\times n}$ is the unweighted adjacency matrix of the given topology, $p_{ij} \in \mathbb{R}$ is the element of the i-th row and j-th column of $\boldsymbol{P}$, representing the probability of node i choosing node j's information, and $\boldsymbol{v} \in \mathbb{R}^n$ is the vector obtained from Program (4.11).

Constraint (4.12b) aims to force the generated transition matrix $\boldsymbol{P}$ to have non-null entries only in the positions where $\boldsymbol{\mathcal{A}}_0$ has 1 as entry to keep the network structure precisely as it was given. It ensures there is no link creation at the generated topology.

Constraint (4.12f) is used to ensure $\boldsymbol{P}$ as a row-stochastic matrix, and Constraint (4.12i) limits its values to be between 0 and 1, a necessary condition to the assumption about its spectrum.

Constraint (4.12g) is the linear matrix inequality version of the second-order cone constraint $\|\boldsymbol{P}^\mathsf{T}\boldsymbol{v} - \boldsymbol{v} + \boldsymbol{P}\boldsymbol{u} - \boldsymbol{u}\|_2 \leq s$ (look at Lobo et al. (1998), Boyd and Vandenberghe (2009) for more about the relationship between semidefinite and second-order cone programming) and it aims to ensure $\boldsymbol{v}^\mathsf{T}$ and $\boldsymbol{u}$ as left and right-eigenvectors of $\boldsymbol{P}$ related to its eigenvalue 1.

Constraint (4.12h) is essential to the correctness of the optimal solution $\boldsymbol{P}^\star$, since it avoids the underlying graph to be disconnected and ensures that it has at least a directed spanning tree in its structure, a necessary condition to the convergence of consensus. This constraint is built as an LMI based on the work of (RAFIEE; BAYEN, 2010), where they tackle a similar problem involving the Laplacian of graphs, yielding

$$\lambda_2(\hat{\boldsymbol{\mathcal{L}}}) > 0 \Leftrightarrow \alpha\mathbf{1}_n\mathbf{1}_n^\mathsf{T} - \nu\boldsymbol{I}_n + \hat{\boldsymbol{\mathcal{L}}} \succeq 0 \tag{4.13}$$

where $\alpha, \nu \in \mathbb{R}$ and $\nu > 0$. The proof for this statement follows (RAFIEE; BAYEN, 2010). Note that, $\boldsymbol{P}$ does not need to be symmetric at all, that is the reason equality (4.12d) is used to ensure that $\hat{\boldsymbol{\mathcal{L}}}$ is symmetric and can be applied on constraint (4.12h) since there is no notion of positiveness for non-Hermitian matrices.

Once matrix $\boldsymbol{P}^\star$ is obtained as solution of Program (4.12), one can find the corresponding adjacency matrix $\boldsymbol{\mathcal{A}}^\star$ directly, since $\boldsymbol{P}^\star$ is just the row-normalised version of $\boldsymbol{\mathcal{A}}^\star$, yielding

$$\boldsymbol{\mathcal{A}}^\star = \operatorname{diag}([\bar{p}_1^\star,\ \bar{p}_2^\star,\ \ldots,\ \bar{p}_n^\star])^{-1}\boldsymbol{P}^\star \tag{4.14}$$

where $\bar{p}_i^\star = \max_j\{p_{ij}^\star\}$ is the maximum value at i-th row of $\boldsymbol{P}^\star$, and $p_{ij}^\star$ is the element in the i-th row and j-th column of $\boldsymbol{P}^\star$. Matrix $\boldsymbol{\mathcal{A}}^\star$ represents the unnormalised adjacency matrix of the network topology that asymptotically drives Equation (4.9) from $\boldsymbol{Z}_0$ to $\boldsymbol{Z}_r$ in finite time, if the underlying graph is primitive, as stated in Proposition 4.2.1.

Figure 4.7 depicts the flowchart of the proposed approach using spectral optimisation regarding a desired spreading distribution, $\boldsymbol{Z}_r$, for each unique information at $\boldsymbol{x}_0$, the initial network topology described by $\boldsymbol{\mathcal{A}}_0$, and the initial spreading distribution of each unique information in $\boldsymbol{x}_0$, represented by $\boldsymbol{Z}_0$.

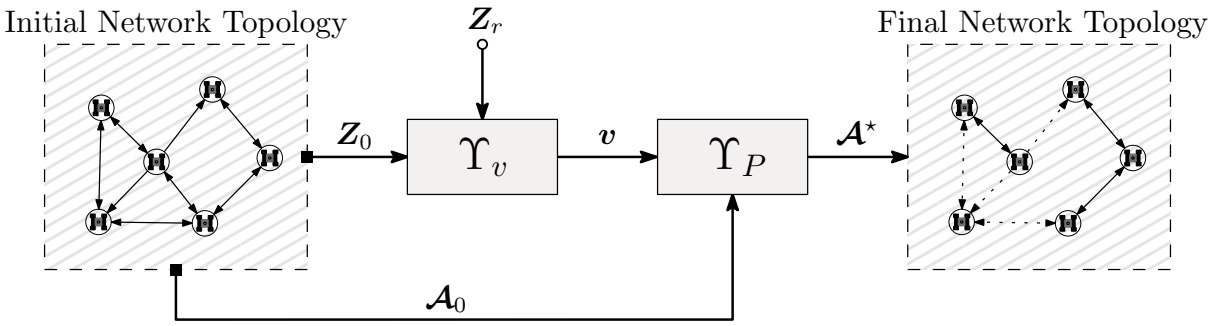


Figure 4.7: Spectral optimisation flowchart for optimal network weight design: Υ_v and Υ_P are the proposed optimisation programs; $\boldsymbol{Z}_0$ and $\boldsymbol{Z}_r$ are the initial and desired spreading distribution for all unique information in $\boldsymbol{x}_0$, respectively; $\boldsymbol{\mathcal{A}}_0$ is the given unweighted adjacency matrix of the underlying network; $\boldsymbol{\mathcal{A}}^\star$ is the optimal adjacency matrix that originates $\boldsymbol{P}^\star$; and $\boldsymbol{v}$ is the leading left-eigenvector of $\boldsymbol{P}^\star$.

Algorithm 2 describes the computational procedure to find the optimal topology $\boldsymbol{\mathcal{A}}^\star$. It is basically the execution of Program (4.11) to find the eigenvector centrality corresponding to $\boldsymbol{Z}_r$, and (4.12) to find the optimum transition matrix corresponding to $\boldsymbol{v}$, followed by Equation (4.14). Note that, this algorithm must be executed offline by a unit with global knowledge, since information about the entire network is necessary to the optimisation programs.

Algorithm 2: Optimal weight design using spectral optimisation

Input: $\boldsymbol{\mathcal{A}}_0,\ \boldsymbol{Z}_0,\ \boldsymbol{Z}_r$
Output: $\boldsymbol{P}^\star,\ \boldsymbol{\mathcal{A}}^\star$

1: $\boldsymbol{v} \leftarrow \Upsilon_v(\boldsymbol{Z}_0, \boldsymbol{Z}_r)$ `// gets the desired eigenvector centrality`
2: $\boldsymbol{P}^\star \leftarrow \Upsilon_P(\boldsymbol{v}, \boldsymbol{\mathcal{A}}_0)$ `// gets the desired transition matrix`
3: $\boldsymbol{\mathcal{A}}^\star \leftarrow \mathrm{diag}([\bar{p}_1^\star,\ \bar{p}_2^\star,\ \ldots,\ \bar{p}_n^\star])^{-1}\boldsymbol{P}^\star$ `// gets the desired adjacency matrix`
4: **return** $\boldsymbol{P}^\star,\ \boldsymbol{\mathcal{A}}^\star$

4.3. Simulations and Results

This section provides numerical examples to evaluate how the proposed network reweighting algorithm influences the DSRCP's convergence in a multi-agent system when changing the network spectrum. A group of simulated agents performs a cooperative task of aggregation[3]: they must select distributively a place where the entire team can meet.

[3]Distributed aggregation is an often used task to evaluate discrete-space consensus approaches, being more explored in (VALENTINI, 2017).

Each meeting point spreads its absolute position in a certain radius, and the agents in this area, receiving that information, can calculate their distance to such a place. Each agent defines its initial information state as the label of the nearest meeting point it received. For simplicity, all agents are under the spreading area of at least one meeting point. It is expected the most "popular" place among the agents to be, in the average, the most chosen one if numerous experiments are performed.

A random geometric graph represents the multi-agent underlying network topology with 20 nodes placed using uniform distribution in the space $[0, 1)$ and 96 links generated regarding the radius parameter of 0.3667, as depicted in Figure 4.8, along with 20 self-loops[4] that are not represented in the figure. Despite random, the graph is ensured to be primitive since it is a necessary condition for the DSRCP's convergence. This kind of topology resembles social and cooperative networks and can describe multi-agent systems accurately in real-world problems, besides other systems over ad hoc networks, concerning their communication topology. The interested reader shall consult (PENROSE, 2003) for more about the theory of random geometric graphs and their applications.

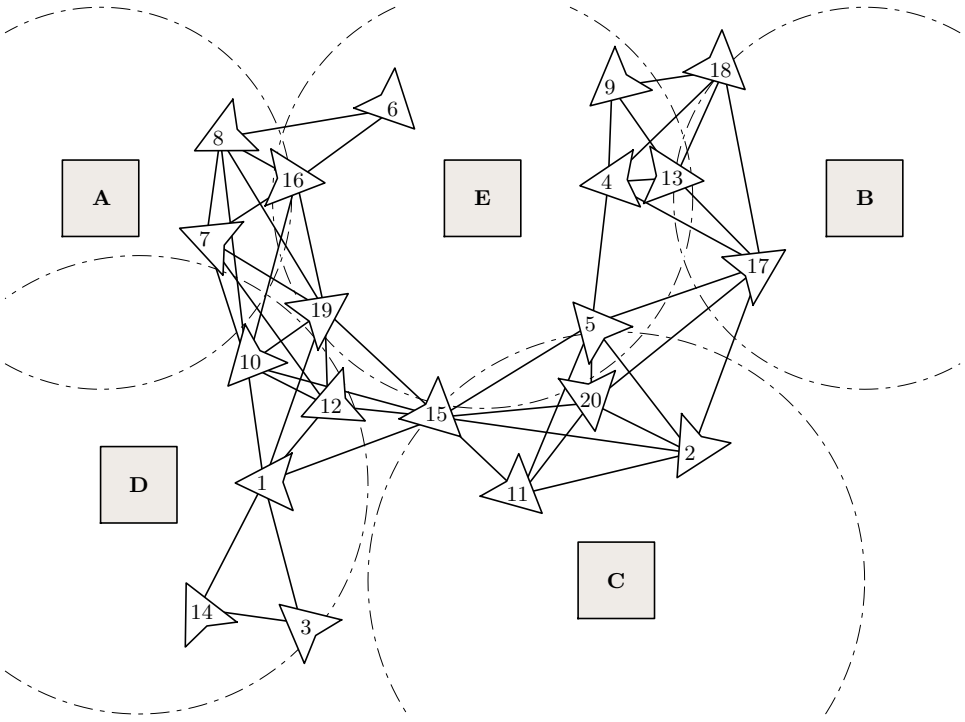


Figure 4.8: Distributed aggregation scenario using a random geometric graph with 20 nodes and 96 links disposed among 5 meeting points. The squares are the possible meeting points, the triangles are the agents pointing to their nearest meeting point, the lines are the communication links between the agents, and the circles represent the diffusion area of each meeting point.

There are five possible distinct meeting points that are displaced over the map in a way that all agents are covered for at least one meeting point and the overlap of their spreading areas is minimum. They are labelled by letters A, B, C, D, and E that represent the countable information to be exchanged. In Figure 4.8, the meeting points are represented by the squares, and the circles around then are their spreading area. Each agent i chooses the label of the nearest meeting point as its initial information state $x_i[0]$.

[4]Self-loops indicate that each node considers its information state when executing the DSRCP.

They are represented in Figure 4.8, as triangles that point in the direction of their nearest meeting point. Nodes 7, and 8 choose place A; nodes 13, 17, and 18 choose place B; nodes 2, 11, 15, and 20 choose place C; nodes 1, 3, 10, 12, and 14 choose place D; nodes 4, 5, 6, 9, 16, and 19 choose place E. Thus, the network initial information state is defined as

$$\boldsymbol{x}_0 = [\mathrm{D},\ \mathrm{C},\ \mathrm{D},\ \mathrm{E},\ \mathrm{E},\ \mathrm{E},\ \mathrm{A},\ \mathrm{A},\ \mathrm{E},\ \mathrm{D},\ \mathrm{C},\ \mathrm{D},\ \mathrm{B},\ \mathrm{D},\ \mathrm{C},\ \mathrm{E},\ \mathrm{B},\ \mathrm{B},\ \mathrm{E},\ \mathrm{C}]^{\mathsf{T}}.$$

The most popular meeting point is E chosen by 6 agents, followed by D chosen by 5 agents. The less popular is A, chosen by only two agents. On the other hand, regarding the network eigenvector centrality, the most influential nodes are the ones with the largest neighbourhood, as depicted in Figure 4.9. According to that, the most influential node in this network is node 15 with 8 neighbours and whose absolute centrality score is 0.0776, followed by nodes 10 and 19 with 7 neighbours and centrality score of 0.0690 each.

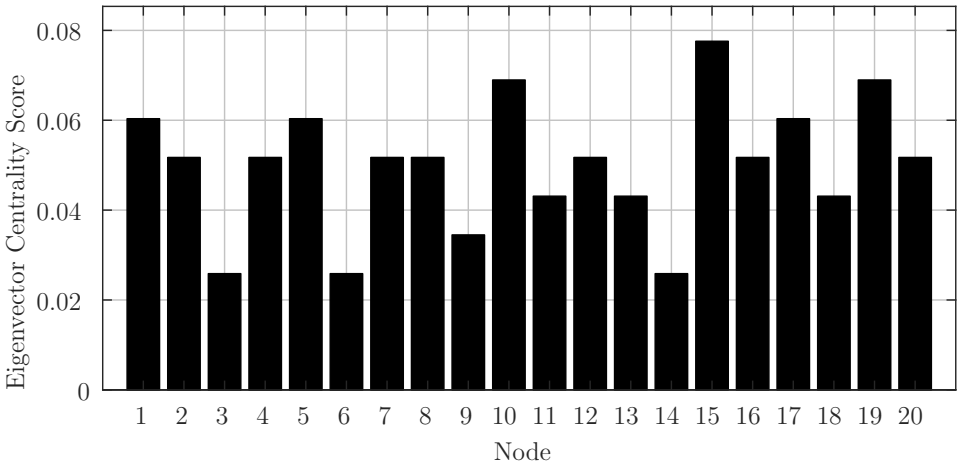


Figure 4.9: Absolute eigenvector centrality score for the given network topology.

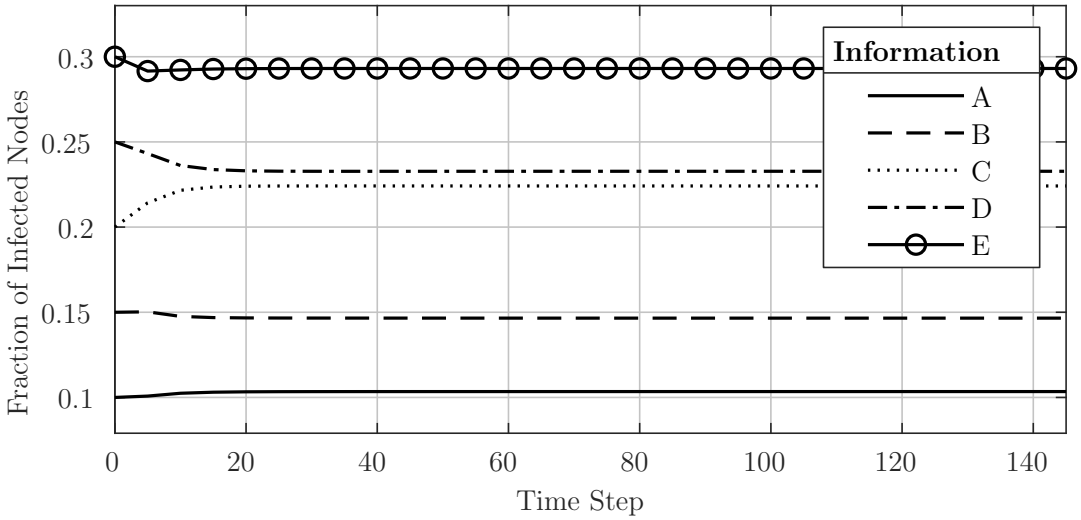
The simulations are organised in three sets: in the first one, the agents perform some iterations of the DSRCP algorithm to reach consensus about the meeting point using the given network topology, then the consensus distribution is analysed for all the meeting points regarding their popularity and the eigenvector centrality; in the second set, the weights of the links are changed offline using the proposed network reweighting approach to increase the consensus convergence to a particular meeting point over the others, and the DSRCP runs over this network to show that the consensus distribution follows the designed distribution; in the third set, the influence of a particular node over the network is changed by increasing its eigenvector centrality through the proposed network reweighting approach executed offline, and it is shown how that affects the consensus convergence.

4.3.1. Reaching Consensus over the Meeting Points

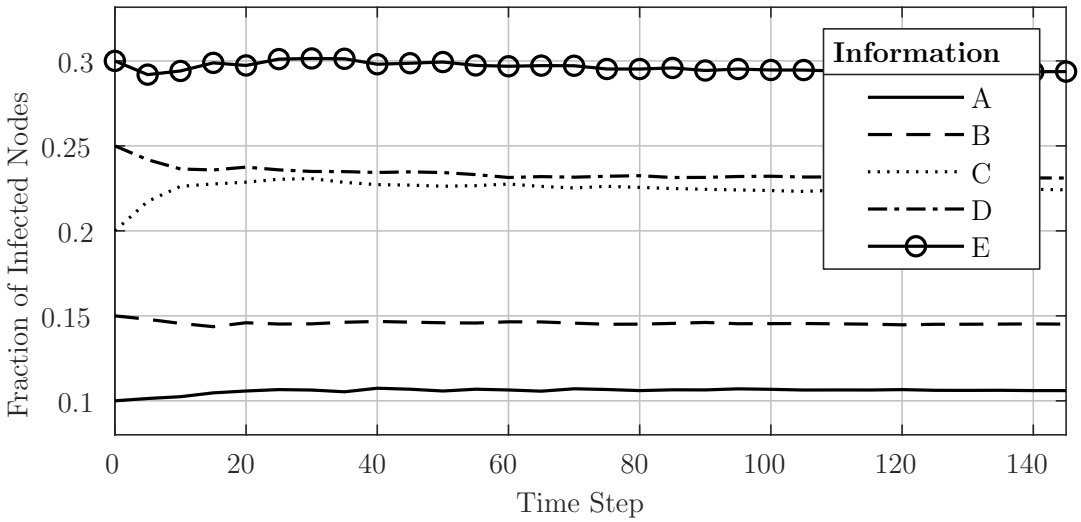
This set of simulations aims to describe the DSRCP's behaviour during the decision-making process of choosing a common meeting point to the agents. Firstly, the DSRCP spreading model is used to predict the consensus distribution of each meeting point. Secondly, 10000 independent experiments are executed, corresponding to the execution of Algorithm 1 with $\overline{l} = 10000$ and $\overline{k} = \infty$. Each experiment is equivalent to every agent executing Equation (4.1) simultaneously until they reach a consensus in any meeting point. Then, the number of times the consensus happens in each meeting point (given by $\boldsymbol{o}$)

regarding the number of experiments is compared with the predicted consensus probability to verify the precision of the spreading model.

By using the eigenvector centrality score and the distribution of the initial infection probabilities, $\boldsymbol{Z}_0$, one can estimate the consensus probabilities for each one of the meeting points, as described in Theorem 4.1. From that, each meeting point has the following consensus probabilities: $\gamma^{\mathrm{A}} = 0.1034,\ \gamma^{\mathrm{B}} = 0.1466,\ \gamma^{\mathrm{C}} = 0.2241,\ \gamma^{\mathrm{D}} = 0.2328,$ and $\gamma^{\mathrm{E}} = 0.2931$. These values are also the steady-state of the expected fraction of infected nodes by each discrete information that represents the meeting points, as depicted in Figure 4.10a. The most likely meeting place to be chosen is E, followed by D, which is in accord with their popularity.



(a)



(b)

Figure 4.10: Expected fraction of infected nodes for each unique information at $\boldsymbol{x}_0$ over time shown in (a), effective fraction of infected nodes for 10000 independent experiments shown in (b).

Figure 4.10b depicts the effective fraction of infected nodes by each unique information in $\boldsymbol{x}_0$ over time, measured from the performed 10000 independent experiments and given by $\boldsymbol{y}[k] = \mathbf{1}_{20}^{\mathsf{T}}\boldsymbol{C}[k]/(10000 \times 20),\ \forall k \in \{0, \ldots, 145\}$. Note as, those trajectories are

pretty similar to the ones describing the expected fraction of infected nodes described in Figure 4.10a that was obtained from the DSRCP spreading model. In Figure 4.11, one can see the error between the expected and the effective fraction of infected nodes observed in the experiments for all information over time. The tiny magnitude of the errors confirms that the DSRCP spreading model is accurate enough to make predictions about the information spreading over time.

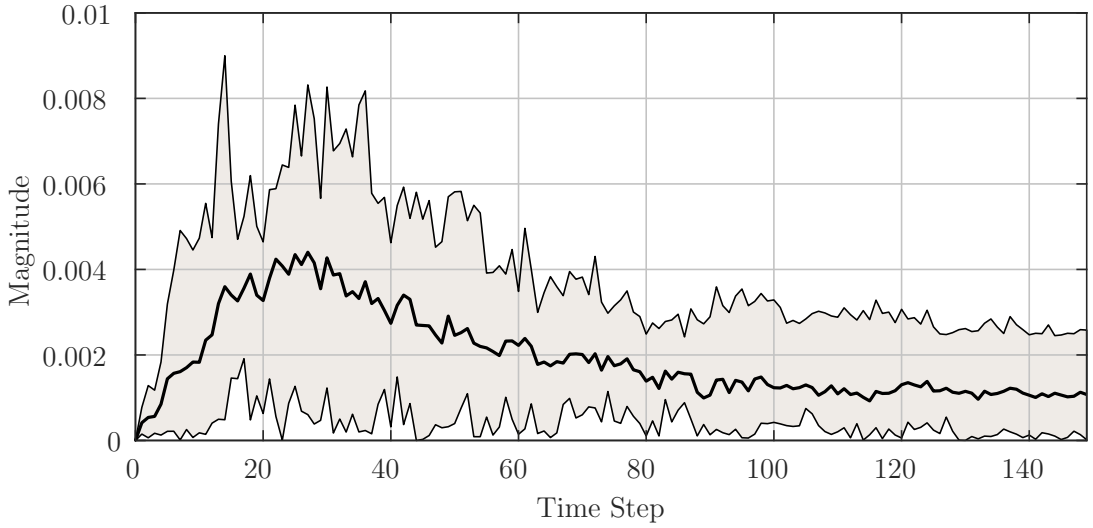


Figure 4.11: Absolute error between the expected and effective fraction of infected nodes obtained from 10000 independent experiments. The shadow indicates the area between the minimum and maximum absolute errors, the line between the extremes is the average error for all information over time.

Figure 4.12a depicts the consensus occurrence for each distinct information, the plus symbol indicates the expected fraction of infected nodes at steady-state for each unique information. One can see the occurrence of consensus and the expected fraction of infected nodes at steady-state match pretty well, and as shown in Figure 4.12b, their match error are significantly small, having in modulo the maximum value of 0.0024 (information A and D), and the minimum value of 0.0001 (information C). That suggests the consensus probabilities are correctly predicted by the DSRCP spreading model. The exact occurrence value for all information and the error between their ratio and the predicted consensus probabilities are also exposed in Table 4.1 for clarification.

Table 4.1: Consensus occurrence of each information obtained from 10000 independent experiments compared to the predicted probability by the DSRCP spreading model for simulation set 1.

Statistics	**Information**				
	A	B	C	D	E
occurrence	1058	1453	2242	2304	2928
occurrence ratio	0.1058	0.1453	0.2242	0.2304	0.2928
predicted probability	0.1034	0.1466	0.2241	0.2328	0.2931
absolute error	0.0024	0.0013	0.0001	0.0024	0.0003

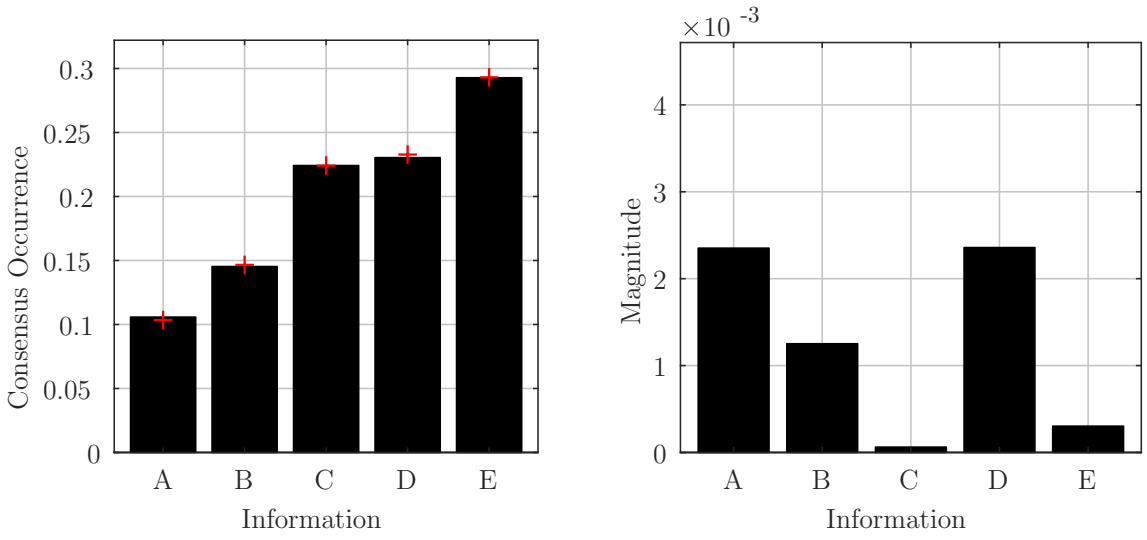


Figure 4.12: Histogram of occurrence of consensus in each distinct information shown in (a), error between the occurrence of consensus and the expected fraction of infected nodes at steady-state by each distinct information shown in (b). The plus symbols indicate the expected fraction of infected nodes at steady-state for each unique information.

4.3.2. Manipulating the Consensus Probability of the Meeting Points

This set of simulations aims to evaluate the effectiveness of the proposed network reweighting algorithm to drive the consensus over a multi-agent system executing the DSRCP. New consensus probabilities are defined for each meeting point, through the offline execution of the proposed semidefinite programs using as input the network of Figure 4.8 and the desired consensus distribution, as described by Algorithm 2. Then, 10000 independent experiments are executed again, as in the previous simulation set, but using as input matrix $\boldsymbol{P}^{\star}$ obtained from Program (4.12). Finally, the ratio between the consensus' occurrence for all the experiments and the total number of experiments is compared with the predicted consensus probabilities given by the DSRCP's spreading model to show their similarity.

The consensus probability of meeting point A (which is the lowest value given the original unweighted topology) increases. Besides that, the consensus probability of all other places also changes – the most "popular" places at the original topology have their probabilities decreased – to show the multi-epidemic feature of the proposed approach. The new consensus probabilities assume the following values: places B, C, and D with 0.0433, place E with 0.1001, and place A with 0.77. That results in the following desired probability distribution for the consensus over the meeting points

$$\boldsymbol{Z}_r = \mathbf{1}_{20}[0.77,\ 0.0433,\ 0.0433,\ 0.0433,\ 0.1001]$$

where $\mathbf{1}_{20}$ is a unity array of order 20.

Figure 4.13a depicts the expected fraction of infected nodes for the topology $\boldsymbol{\mathcal{A}}^{\star}$ obtained from the optimisation program. This topology has no new links when compared to the initial topology, just its links' weights were changed. Note the substantial increase in the strength of information A when compared to the initial topology: its expected

fraction of infected nodes goes from 0.1034 in $\boldsymbol{\mathcal{A}}_0$ to 0.77 in the final topology. Also, information E's expected fraction of infected nodes goes from 0.2931 in $\boldsymbol{\mathcal{A}}_0$ to 0.1001. All other information have their expected fraction of infected nodes stabilised at 0.0433, as designed above.

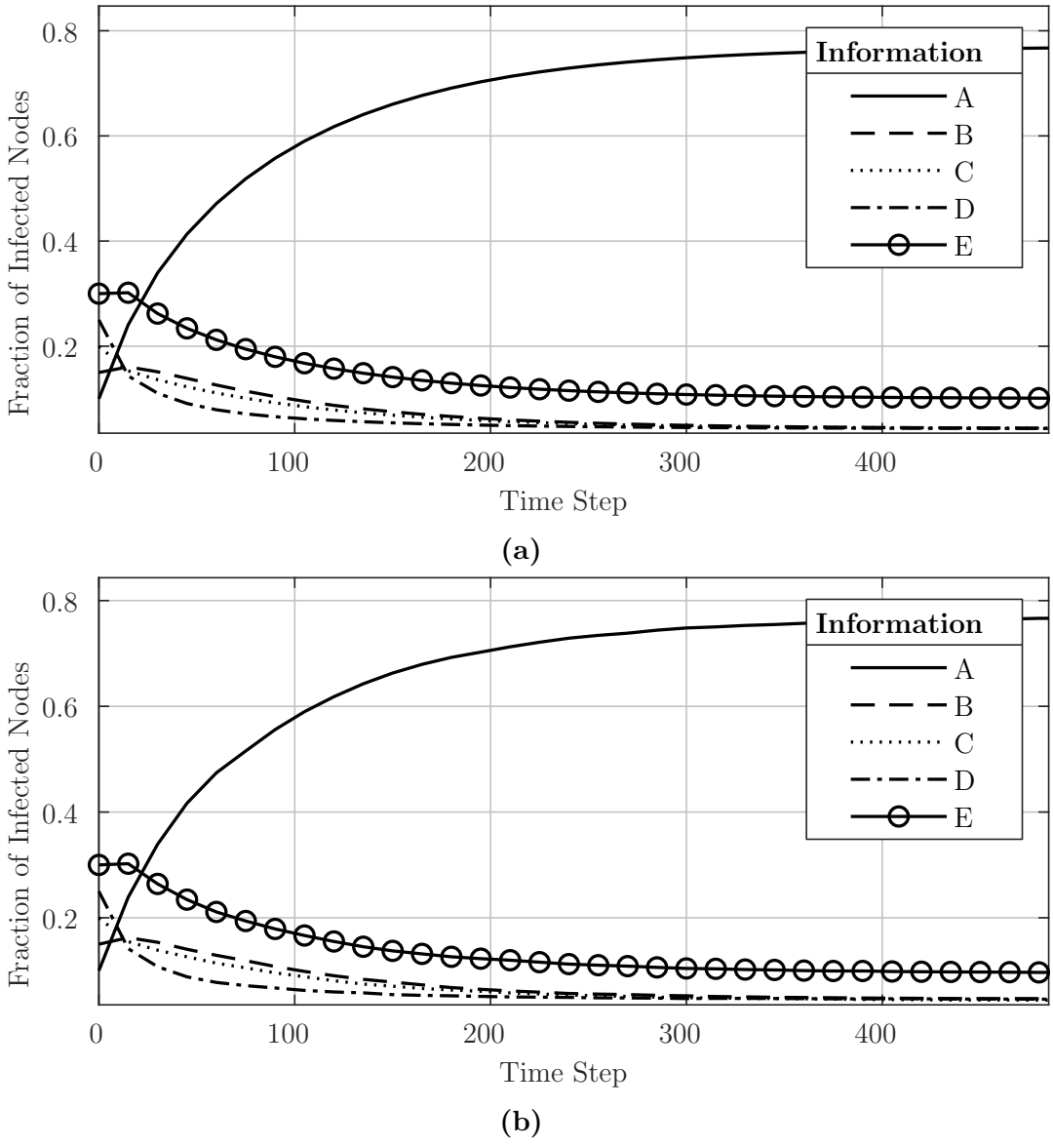


Figure 4.13: Expected fraction of infected nodes for each information at $\boldsymbol{x}_0$ over time shown in (a), effective fraction of infected nodes for 10000 independent experiments shown in (b). In both cases the topology used is $\boldsymbol{\mathcal{A}}^\star$.

Figure 4.13b shows the effective fraction of infected nodes by each information in $\boldsymbol{x}_0$ obtained from the execution of 10000 independent experiments, in the same way as performed in the previous simulation set, except it uses $\boldsymbol{P}^\star$ as input for Algorithm 1. One can see, there is no visible difference between the trajectories of both expected and effective fraction of infected nodes. As shown in Figure 4.14, their absolute error over time remains below 0.005 all the time for every information, what indicates with high confidence the DSRCP spreading model correctly describes the evolution of infection probabilities for all unique information in $\boldsymbol{x}_0$.

The change in the expected fraction of infected nodes at the steady-state for each

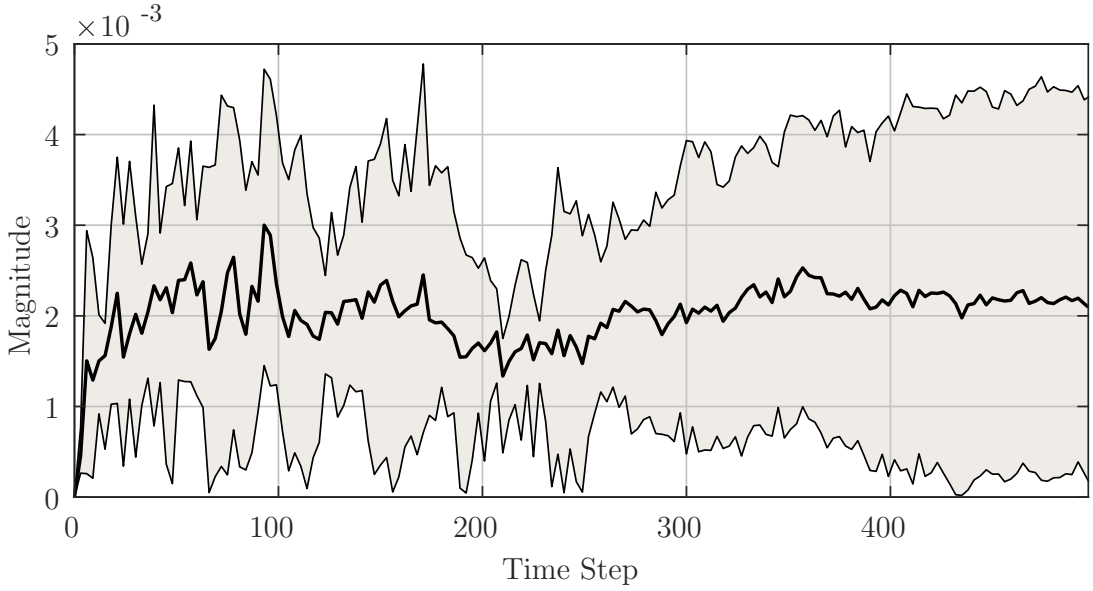


Figure 4.14: Absolute error between the expected and effective fraction of infected nodes obtained from 10000 independent experiments. The shadow indicates the area between the minimum and maximum absolute errors, the line between the extremes is the average error for all information over time.

information should also impact the eigenvector centrality score of every node, since the first is the result of a convex combination of the second, as shown in Section 4.1.3. In fact, the nodes' centrality score changed according to the designed consensus probabilities for each information, as detailed in the parallel with the original scores plotted in Figure 4.16. Note as the score of nodes 7 and 8 massively increased, as these nodes are "infected" by information A at the beginning. Also, be aware that $v_7 z_7^{\mathrm{A}}[0] + v_8 z_8^{\mathrm{A}}[0] = 0.3176 + 0.4524 = 0.77$, as stated before.

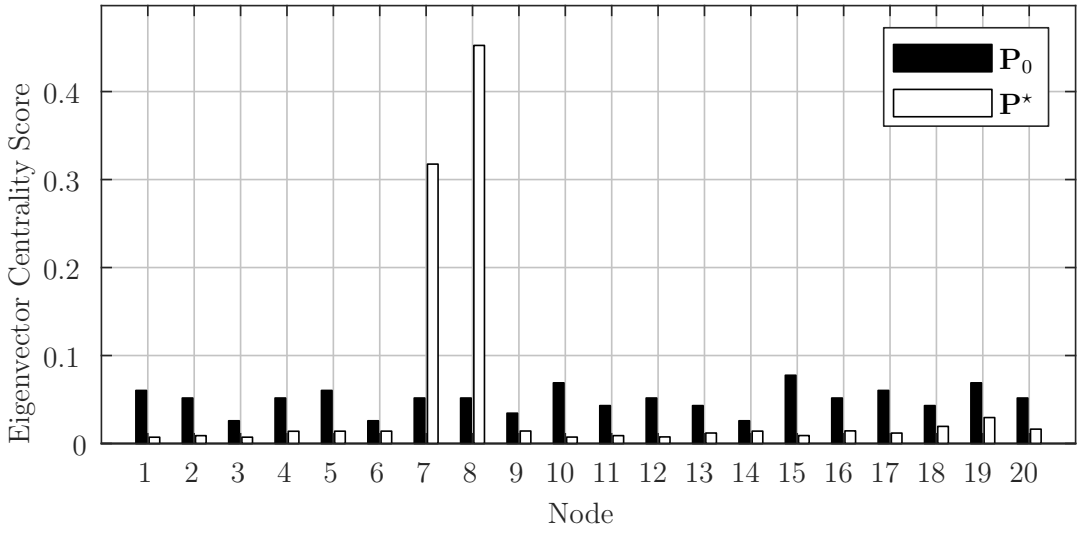


Figure 4.15: Absolute eigenvector centrality score for the given topology ($\boldsymbol{P}_0$) versus the obtained from the generated topology ($\boldsymbol{P}^\star$).

As shown in Section 4.1.4, the expected fraction of infected nodes for each information indicates the probability of such information be prevalent when consensus is reached.

Accordingly, the values at steady-state presented in Figure 4.13, should represent the probability of consensus in each information. This hypothesis is evaluated through the execution of Algorithm 1 with $\overline{l} = 10000$, $\overline{k} = \infty$, and using $\boldsymbol{P}^{\star}$, obtained from Algorithm 2, as input. Figure 4.16a depicts the occurrence of consensus in each information, and Figure 4.16b portrays the error between the expected fraction of infected nodes and the occurrence of consensus in each information. Note that, the error between the expected fraction of infected nodes and the consensus occurrence in each information is tiny: the absolute minimum error is 0.0005 for place C, and the maximum is 0.0046 for place E, what indicates the success of the proposed network reweighting approach to drive the consensus in the given topology. Table 4.2 summarises those results.

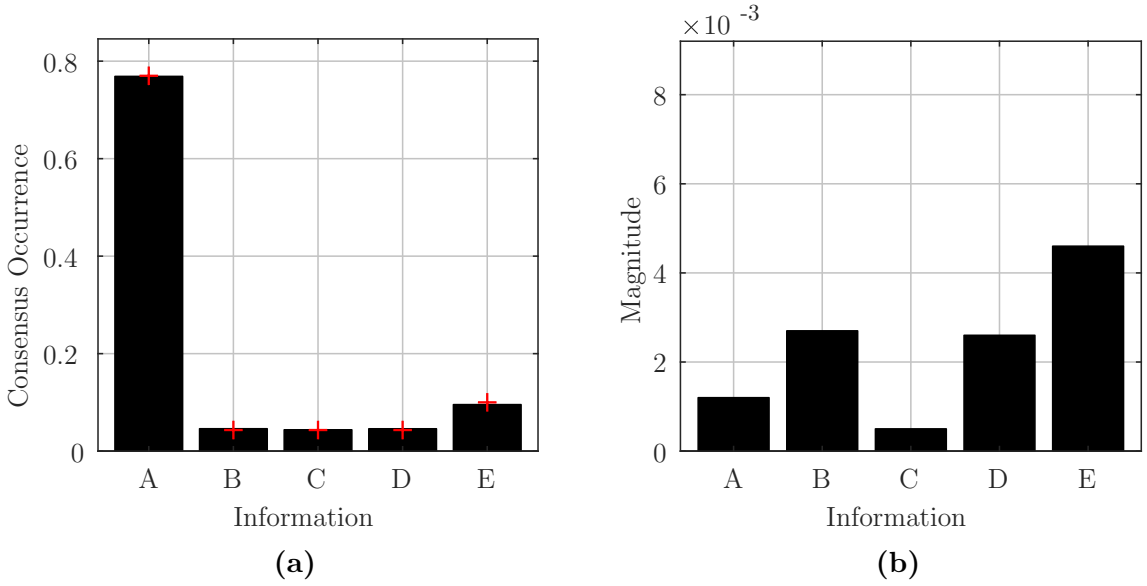


Figure 4.16: Histogram of occurrence of consensus in each distinct information shown in (a), error between the occurrence of consensus and the expected fraction of infected nodes at steady-state by each distinct information shown in (b). The plus symbol indicates the expected fraction of infected nodes at steady-state for each unique information.

Table 4.2: Consensus occurrence of each information obtained from 10000 independent experiments compared to the predicted probability by the DSRCP spreading model for simulation set 2.

Statistics	**Information**				
	A	B	C	D	E
occurrence	7688	460	438	459	955
occurrence ratio	0.7688	0.046	0.0438	0.0459	0.0955
predicted probability	0.77	0.0433	0.0433	0.0433	0.1001
absolute error	0.0012	0.0027	0.0005	0.0026	0.0046

One can see in Figure 4.17, the heatmap of the initial adjacency matrix (a) and the heatmap of the generated adjacency matrix (b). The colours represent the weight of each link: the darker, the higher is the weight, the lighter, the lower is it. As the weights belong to the range $[0, 1]$, the optimisation program finds the best network topology to

fits the desired eigenvector centrality and the desired spreading distribution through the weakening of some links, while it keeps others with their original strength. Hence, there is no addition of new links to the given topology. This idea of using weak links to change the network behaviour is extensively explored in complex systems theory as stated for example in (CSERMELY, 2009), and this is a practical example of the use of weak links to change an intrinsic property of a networked system.

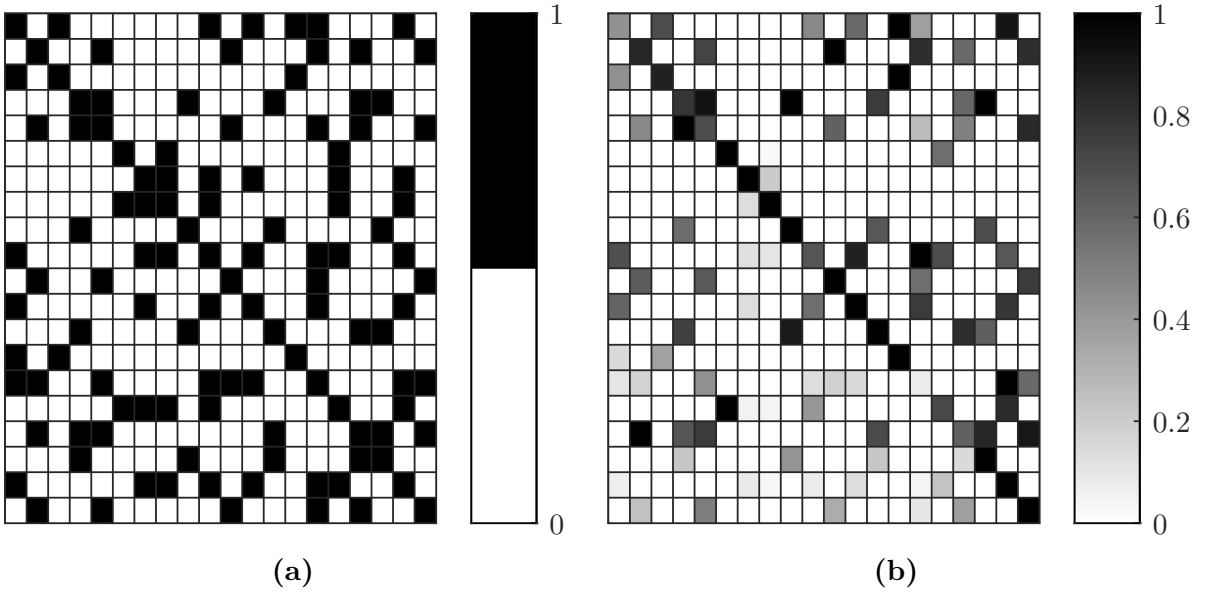


Figure 4.17: Heatmap of the initial adjacency matrix $\mathcal{A}_0$ shown in (a), and of the generated adjacency matrix $\mathcal{A}^\star$ shown in (b). The colours represent the weight of each link: the darker, the higher is the weight, the lighter, the lower is it.

4.3.3. Manipulating the Agents Influence over the Network

In the previous section, the consensus probability of the meeting points is settled to the desired value, changing, by consequence, the eigenvector centrality of the nodes that initially have preferred such meeting points. In this section, the social power of each agent is manipulated directly, independently of their initial information, through the execution of Program (4.12) using $\boldsymbol{v}$ as the desired eigenvector centrality. Then, 10000 independent experiments are performed following the same scheme of the previous simulation sets to evaluate if the system behaves as desired.

By changing the eigenvector centrality score of each agent in the network, one can attribute more importance for some nodes and less to others. This capability is especially attractive when there are agents whose knowledge or information is more (or less) reliable than from others. Thus, by changing those agents' eigenvector centrality score, one can manipulate their social power, allowing trustable agents to be more influential than untrustable ones, for example.

In this simulation set, the scenario is the same described in Figure 4.8, but the eigenvector centrality score of agent 3 (whose value is the lowest among other agents in the original topology) increases to show its influence in the consensus over the meeting points. Also, the influence of nodes 10, 15, and 19 (that are the highest in the original topology) decreases. The desired eigenvector centrality score assumes the following values:

$v_{10} = 0.001,\ v_{15} = 0.001,\ v_{19} = 0.001$, and $v_3 = 0.2385$. All other nodes' centrality score is the same as presented in the original topology.

Figure 4.18 compares the eigenvector centrality score of the original topology against the ones from $\boldsymbol{P}^\star$. Note that, the increase in node 3's score is exactly the sum of the difference between the original scores of nodes 10, 15, 19 and their respective desired scores, i.e. $v_{10} = 0.069 - 0.068 = 0.001$, $v_{15} = 0.0776 - 0.0766 = 0.001$, $v_{19} = 0.069 - 0.068 = 0.001$, and $v_3 = 0.0259 + 2 \times 0.068 + 0.0766 = 0.2385$.

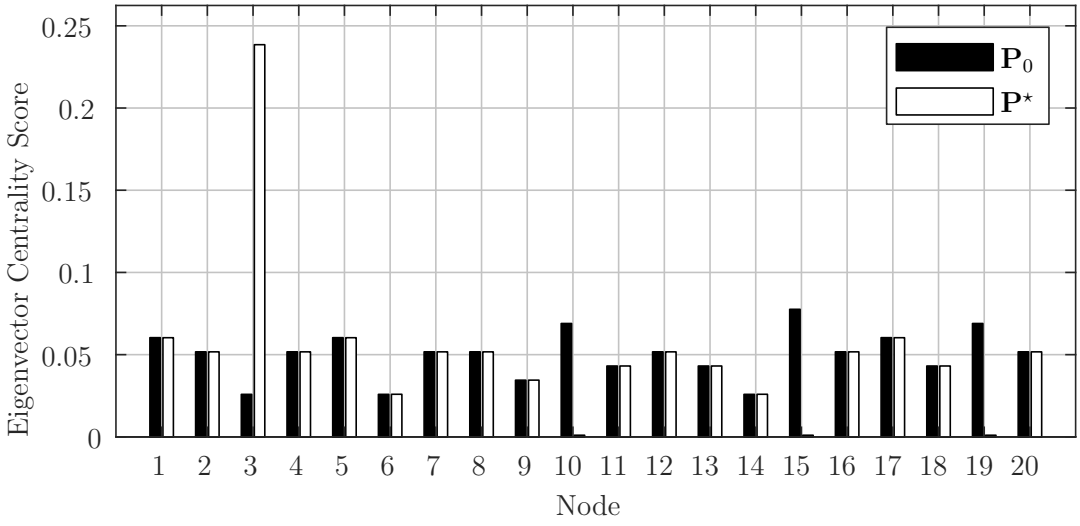


Figure 4.18: Absolute eigenvector centrality score for the given topology ($\boldsymbol{P}_0$) versus the obtained from the generated topology ($\boldsymbol{P}^\star$).

The new eigenvector centrality scores indicate that the meeting point supported by node 3 has, at least[5], 23.85% of chance to be prevalent for all the group. In the same way, the meeting points supported by agents 10, 15, and 19 have, at least, 0.1% of chance to be chosen by all other agents. Those values are the contribution of each agent in the spreading of their initial information.

By using the new eigenvector centrality score and the initial distribution of information, $\boldsymbol{Z}_0$, one can estimate the consensus probability for all the meeting points, yielding $\gamma^{\mathrm{A}} = 0.1034,\ \gamma^{\mathrm{B}} = 0.1465,\ \gamma^{\mathrm{C}} = 0.1475,\ \gamma^{\mathrm{D}} = 0.3774,\ \gamma^{\mathrm{E}} = 0.2251$. One can see how the consensus probabilities changed when compared to the values of the original topology. In fact, there is an increase of 0.1446 in the consensus probability of meeting point D, which is supported by node 3 at the beginning. This value is precisely the amount of increase in the eigenvector centrality score of node 3. Also, there is a decrease of the consensus probabilities of meeting points B (of 0.0001), C (of 0.0766), and E (of 0.068). Those values for information C and E are the amount of decreasing in the centrality scores of nodes 15 and 19, respectively. However, the decreasing in the consensus probability of meeting point B has no clear origin, although it is likely to be related to the decreasing of node 15's influence, which is a critical node and, consequently, impacts the information flow between the two partitions of the network.

Figure 4.19a depicts the consensus occurrence for 10000 independent experiments performed by executing Algorithm 1 using the optimised topology $\boldsymbol{P}^\star$ as input. One can see the consensus occurrence match pretty well the probabilities calculated by the DSRCP spreading model. As shown in Figure 4.19b, the maximum error is of 0.0102 (for meeting

[5]If only agent 3 has such information at beginning, the probability of consensus over that is 23.85%.

point C), and the minimum error is of 0.0016 (for meeting point D). Table 4.3 summarises such results.

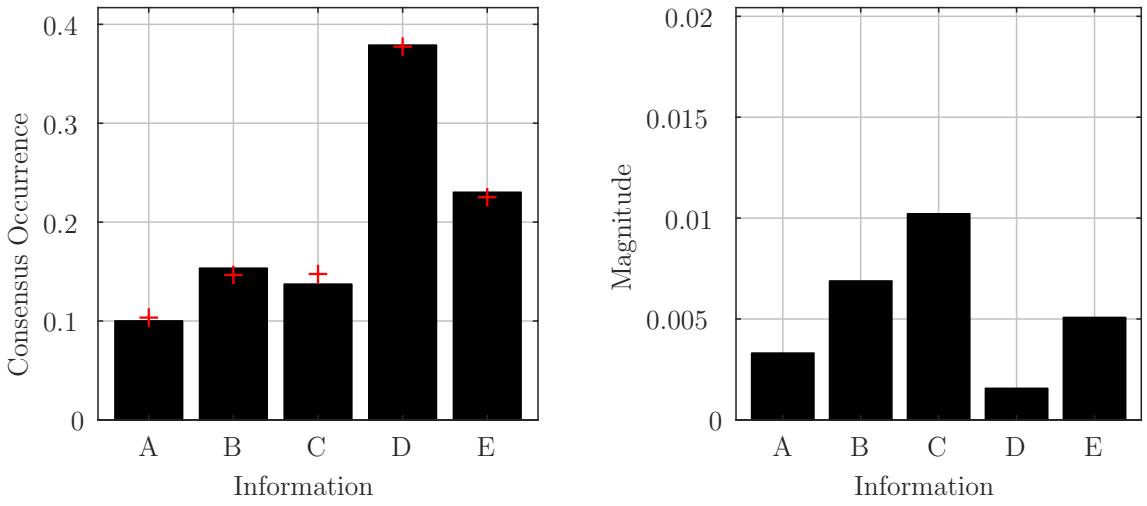


Figure 4.19: Histogram of occurrence of consensus in each distinct information shown in (a), error between the occurrence of consensus and the expected fraction of infected nodes at steady-state by each distinct information shown in (b). The plus symbol indicates the predicted consensus probability for each unique information.

Table 4.3: Consensus occurrence of each information obtained from 10000 independent experiments compared to the predicted probability by the DSRCP spreading model for simulation set 3.

Statistics	**Information**				
	A	B	C	D	E
occurrence	1001	1534	1373	3790	2302
occurrence ratio	0.1001	0.1534	0.1373	0.3790	0.2302
predicted probability	0.1034	0.1465	0.1475	0.3774	0.2251
absolute error	0.0033	0.0069	0.0102	0.0016	0.0051

Figure 4.20 portrays the comparison between the original adjacency matrix, $\boldsymbol{\mathcal{A}}_0$, and the one obtained from the optimisation program, $\boldsymbol{\mathcal{A}}^\star$. Note as the influence of node 15 vanished almost completely in $\boldsymbol{\mathcal{A}}^\star$ by looking at the 15th column of matrix described by Figure 4.20b. The only considerable weights are on nodes 10 and 19, that also had their eigenvector scores decreased. Also, one can see the decrease in the influence of node 3's neighbours over it by looking at the 3rd row of $\boldsymbol{\mathcal{A}}^\star$, in Figure 4.20b.

4.4. Summary

This chapter has presented the most significant contribution of this thesis to the field of network control for information spreading manipulation in multi-robot systems: an offline algorithm for network weight design based on semidefinite programming that can

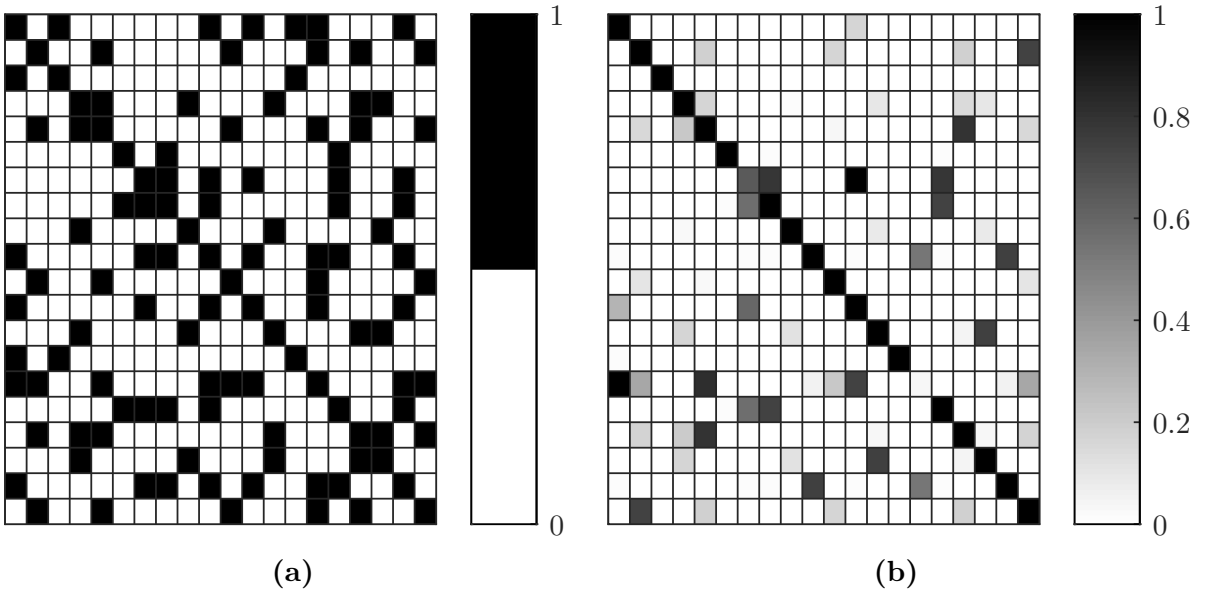


Figure 4.20: Heatmap of the initial adjacency matrix $\boldsymbol{\mathcal{A}}_0$ shown in (a), and of the generated adjacency matrix $\boldsymbol{\mathcal{A}}^{\star}$ shown in (b). The colours represent the weight of each link: the darker, the higher is the weight, the lighter, the lower is it.

drive the convergence of a discrete-space consensus protocol through the manipulation of the network weights. Also, it has presented a new discrete-space consensus protocol, named Discrete-Space Random Consensus Protocol (DSRCP), that is driven by the network topology and can solve the consensus decision-making problem in a multi-robot system using only local information. The epidemiologic machinery was used to model the DSRCP spreading dynamics, and correlate that with the network topology. Such a model was extended to deal with the maximum number of distinct information in the network at once. Extensive simulations have shown that the proposed spreading model is precisely enough and can calculate the infection probability of all given information be in each node of the network at any time step. Also, the proposed network reweighting approach can drive the consensus for all information at the initial state to any desired suitable distribution if the initial topology meets the DSRCP convergence requirements.

Chapter 5

Final Remarks

The network topology can determine directly both, the information flow and the information spreading capability of an interconnected system, allowing one to drive the cooperative behaviours underneath such systems directly through its manipulation. Concerning that, this work has tackled two main aspects of the network topology control in multi-robot systems: topology control for connectivity maintenance and topology control for information spreading manipulation. Both aspects were explored independently in the context of autonomous agents solving tasks cooperatively since this kind of scenario is tremendously dependent on the communication topology characteristics. The presented approaches were individually evaluated in two distinct scenarios: i) a continuous-space consensus problem for optimisation of the sensing areas in a multi-robot system; ii) a discrete-space consensus problem of distributed aggregation in a multi-robot team. Both scenarios share the same core: distributed cooperation among autonomous individuals.

Regarding the thesis' objectives in the field of topology control for connectivity maintenance, it has presented an extension to a former topology control approach used to increase the connectivity of a network of mobile robots using only local information allowing that to deal with wireless signal strength variations and collision avoidance among the agents during the process of increasing the network connectivity. The use of readings from signal strength indicator in wireless networks allows the robots to perceive the environment changes that could compromise their link health and allows them to take actions that could improve those values in order to keep the network connected. The adopted approach uses the solution of a distributed version of the TSP solved for each robot of the system, using their 2-hop local information that can be easily obtained independently of the network size. Such an approach ensures, at the long time running, that the topology is at least bi-connected, being tolerant to failures in until one robot for sure. Besides that, the collision avoidance strategy applied to the connectivity control of the proposed approach is fully decentralised and ensure no collisions among any robots of a multi-robot team. It is formulated as a non-convex problem solved distributively by an optimisation program based on distributed MPC and SCP. Simulations and practical experiments have shown that the proposed extensions improved the former approaches, allowing them to be more efficiently applied on cooperative problems.

In the field of topology control for information spreading manipulation, this thesis has presented a new approach based on semidefinite programming and spectral optimisation to manipulate the convergence of a random discrete-space consensus protocol through the reweighting of given network topology, without adding any new links. The new random discrete-space consensus protocol was also a contribution, it was based on the synchronous voter model, but has its dynamics modelled with tools of mathematical epidemiology and Markov chains. Its dynamical model relates the information spreading directly with some

network properties, such as the eigenvector centrality score. By using such a model, this work has shown the convergence conditions and the equilibrium points of the proposed consensus protocol. Also, through this model, the optimisation algorithm designs optimal weights for the network links in a way that the whole system can reach the desired information spreading. This algorithm ensures that every proper information distribution can be reached since the topology complies with the consensus convergence requirements. Extensive simulations have shown the efficiency of the proposed reweighting approach, as well as, the precision of the dynamical spreading model proposed to the random discrete-space consensus protocol.

The results of this work apply primarily, but not exclusively, for multi-robot systems that follow some cooperative strategy to solve tasks in a decentralised manner. The tasks used in each case – optimisation of multi-robot sensing area and distributed aggregation over multi-agents – were used mainly as a study case for the proposed strategies. One can apply the ideas presented by this thesis in almost any networked system in which individuals must perform information sharing and follow cooperative protocols to solve a particular task. One of these possible applications is, for instance, social networks whose dynamics are mainly driven by the interaction between autonomous agents (their users, generally humans) through information sharing. However, the approaches presented both for maintaining network connectivity and for handling the information dissemination have limitations and should, therefore, be improved according to the requirement of each task. Next sections present the technical productions resultant of this work, the main drawbacks of the proposed approaches, and possible research lines as suggestions of future works that could solve some of those drawbacks.

5.1. Technical Production

This work has produced as a direct outcome the following technical articles submitted (some published) in conferences, symposium, and journals in the interest area:

- CARVALHO, S.; ALMEIDA, L.; MORENO, U. Distributed connectivity management in networks of multiple robots in area coverage tasks. In: **8º INForum - Simpósio de Informática**. Lisbon, Portugal: IST Lisboa, 2016.
- CARVALHO, S.; PINTO, L.; ALMEIDA, L.; MORENO, U. Improving robustness of robotic networks using consensus and wireless signal strength. **IFAC-PapersOnLine**, v. 49, n. 30, p. 337–342, Nov 2016. 4th IFAC Symposium on Telematics Application, Porto Alegre, Brazil.
- RAMALHO, G. M.; CARVALHO, S. R.; FINARDI, E. C.; MORENO, U. F. Trajectory optimization using sequential convex programming with collision avoidance. **Journal of Control, Automation and Electrical Systems**, Springer US, v. 29, n. 3, p. 318–327, Jun 2018.
- SALEM, F.; TCHILIAN, R.; CARVALHO, S.; MORENO, U. Opinion dynamics over a finite set in cooperative multi-robot systems: An asynchronous gossip-based consensus approach. In: **XXIII Congresso Brasileiro de Automática**. Porto Alegre, RS, Brazil: Galoá, 2020.
- CARVALHO, S.; MORENO, U. Optimal design for manipulation of random consensus over discrete information in networked systems. Submitted to the Journal of The Franklin Institute. 2020.

5.2. Limitations

The approaches presented in this work have the following drawbacks concerning the strategies used to solve each one of the research objectives:

- The proposed connectivity control to increase the network connectivity does not take into account any environmental obstacle, despite its capability to avoid collision among the robots;
- Due to the choice of use bi-connected topologies to increase the network robustness to faults in its nodes, this network does not assuredly tolerate faults in more than one robot at once;
- During increased network connectivity, the approach to maintaining connectivity cannot guarantee tolerance to any failures in robots due to process uncertainty;
- The approximations performed by the connectivity control over the collision avoidance constraints causes tiny violations of them;
- The proposed algorithm for network weight design must operate offline using the entire network topology as input, i.e. global knowledge is necessary;
- The computational cost of the proposal for designing the network weights is at least $\mathcal{O}(|\mathcal{E}|)$ for the number of connections in the network, which limits the size of the network according to the available computational power and restricts its applications to certain types of networked systems;
- The spreading model for the proposed random consensus protocol does not contemplate the information that is not present in the initial state of the network and, also, the empty information state (i.e. the absence of information);
- The proposed algorithm for network weight design cannot deal with time-varying topologies.

5.3. Future Works

In line with the main drawbacks of the proposed approaches, there are possible lines of research that look promising for improving them, for example:

- Adapt connectivity control to dynamically detect the presence of obstacles in the environment, so that the topology built at the logical level takes into account the presence of obstacles in the environment that may cause disconnections or make the planned structure unfeasible;
- Increase the connectivity of the network to orders higher than bi-connectivity, resulting in a network topology that is assuredly tolerant to simultaneous faults in more than one robot;
- Smooth the approximations of the collision avoidance constraints or use a non-convex formulation for the connectivity control that allows the constraints to be embedded in their original form without jeopardising the solution of the problem;
- Decentralisation of the proposed approach for network weight design in order to use only local information and to reduce its computational complexity, improving its applicability to more domains;
- Add to the dynamical spreading model of the DSRCP the capability to deal with the absence of information, and the capacity of modelling the forgetfulness or loss of interest which would extend this approach to handle social and other more complex networked systems;
- Allow the proposed approach for network weight design to handle time-varying

topologies, by using other optimisation strategies instead of the spectral optimisation that would not use the network spectrum directly or that could estimate such values from local information;
- Apply the network reweighting strategy in social networks, aiming the control of human social interaction, through the manipulation of their social power which is directly related to the network spectrum, as shown in this thesis.

5.4. Acknowledgement

This work was partially supported by the Brazilian agencies Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) under the grant numbers 207650/2015-2 and 140219/2019-6; and Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) under the grant number 88882.182547/2018-01.

References

ACEMOGLU, D.; OZDAGLAR, A. Opinion dynamics and learning in social networks. **Dynamic Games and Applications**, v. 1, n. 1, p. 3–49, Oct 2011. (cit. on p. 3)

ACEMOGLU, D.; OZDAGLAR, A.; PARANDEHGHEIBI, A. Spread of (mis) information in social networks. **Games and Economic Behavior**, v. 70, n. 2, p. 194–227, Nov 2010. (cit. on p. 3)

ADEWUMI, O. G.; DJOUANI, K.; KURIEN, A. M. Rssi based indoor and outdoor distance estimation for localization in wsn. In: **2013 IEEE International Conference on Industrial Technology**. Cape Town, South Africa: IEEE, 2013. p. 1534–1539. (cit. on p. 25)

ARANDA, M.; ARAGÜÉS, R.; LÓPEZ-NICOLÁS, G.; SAGÜÉS, C. Connectivity-preserving formation stabilization of unicycles in local coordinates using minimum spanning tree. In: **2016 American Control Conference**. Boston, MA, USA: IEEE, 2016. p. 1968–1974. (cit. on p. 4)

ASAVATHIRATHAM, C.; ROY, S.; LESIEUTRE, B.; VERGHESE, G. The influence model. **IEEE Control Systems Magazine**, v. 21, n. 6, p. 52–64, Dec 2001. (cit. on p. 4)

BAILEY, N. T. **The Mathematical Theory of Infectious Diseases**. 2nd. ed. London, UK: Griffin, 1975. (Mathematics in Medicine Series). (cit. on p. 17)

BANISCH, S.; LIMA, R. Markov chain aggregation for simple agent-based models on symmetric networks: the voter model. **Advances in Complex Systems**, v. 18, n. 3 & 4, p. 1550009–23, Jun 2015. (cit. on p. 4)

BANISCH, S.; LIMA, R.; ARAÚJO, T. Agent based models and opinion dynamics as markov chains. **Social Networks**, v. 34, n. 4, p. 549–561, Oct 2012. (cit. on p. 4)

BARAS, J. S.; HOVARESHTI, P. Efficient and robust communication topologies for distributed decision making in networked systems. In: **Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference**. Shanghai, China: IEEE, 2009. p. 3751–3756. (cit. on p. 15)

BONACICH, P. Factoring and weighting approaches to status scores and clique identification. **The Journal of Mathematical Sociology**, v. 2, n. 1, p. 113–120, Aug

1972. (cit. on pp. 15 and 46)

BOYD, S.; GHOSH, A.; PRABHAKAR, B.; SHAH, D. Randomized gossip algorithms. **IEEE Transactions on Information Theory**, IEEE, v. 52, n. 6, p. 2508–2530, Jun 2006. (cit. on pp. 3 and 16)

BOYD, S.; VANDENBERGHE, L. **Convex Optimization**. 7th. ed. New York, NY, USA: Cambridge University Press, 2009. (cit. on p. 56)

BUTTERFIELD, J.; DANTU, K.; GERKEY, B.; JENKINS, O. C.; SUKHATME, G. S. Autonomous biconnected networks of mobile robots. In: **6th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks and Workshops**. Berlin, Germany: IEEE, 2008. p. 640–646. (cit. on p. 5)

CAMACHO, E. F.; ALBA, C. B. **Model Predictive Control**. 2nd. ed. London, UK: Springer-Verlag London, 2007. (Advanced Textbooks in Control and Signal Processing). (cit. on p. 28)

CAMPONOGARA, E.; JIA, D.; KROGH, B. H.; TALUKDAR, S. Distributed model predictive control. **IEEE Control Systems**, v. 22, p. 44–52, 2002. (cit. on p. 28)

CARVALHO, S.; ALMEIDA, L.; MORENO, U. Distributed connectivity management in networks of multiple robots in area coverage tasks. In: **8º INForum - Simpósio de Informática**. Lisbon, Portugal: IST Lisboa, 2016. (cit. on p. 71)

CARVALHO, S.; MORENO, U. Optimal design for manipulation of random consensus over discrete information in networked systems. Submitted to the Journal of The Franklin Institute. 2020. (cit. on p. 71)

CARVALHO, S.; PINTO, L.; ALMEIDA, L.; MORENO, U. Improving robustness of robotic networks using consensus and wireless signal strength. **IFAC-PapersOnLine**, v. 49, n. 30, p. 337–342, Nov 2016. 4th IFAC Symposium on Telematics Application, Porto Alegre, Brazil. (cit. on p. 71)

CARVALHO, S. R. D. d. **Controle de Topologia em Redes de Robôs Móveis Cooperativos Utilizando Consenso**. Dissertação (Mestrado) — Federal University of Santa Catarina, Florianópolis, SC, Brazil, Jul 2015. (cit. on pp. 9 and 24)

CARVALHO, S. R. D. de; CORREIA, F. L. d. B.; MORENO, U. F. Topology control for connectivity maintenance in cooperative mobile robot networks. **IFAC-PapersOnLine**, v. 48, n. 19, p. 280–285, Aug 2015. 11th IFAC Symposium on Robot Control, Salvador, Brazil. (cit. on pp. 5, 9, and 24)

CARVALHO, S. R. D. de; MORENO, U. F. Controle de topologia para a manutenção da conectividade em sistemas multi-robôs utilizando consenso e controle preditivo. In: **XII Simpósio Brasileiro de Automação Inteligente**. Natal, RN, Brazil: SBA, 2015. p. 1023–1028. (cit. on pp. 9 and 24)

CASTEIGTS, A.; ALBERT, J.; CHAUMETTE, S.; NAYAK, A.; STOJMENOVIC, I. Biconnecting a network of mobile robots using virtual angular forces. In: **72nd IEEE**

Vehicular Technology Conference. Ottawa, ON, Canada: IEEE, 2010. (cit. on p. 5)

CERAGIOLI, F.; FRASCA, P. Consensus and disagreement: The role of quantized behaviors in opinion dynamics. **SIAM Journal on Control and Optimization**, v. 56, n. 2, p. 1058–1080, Mar 2018. (cit. on p. 4)

CHEN, D.-B.; GAO, H. An improved adaptive model for information recommending and spreading. **Chinese Physics Letters**, IOP Publishing, v. 29, n. 4, p. 048901–4, Apr 2012. (cit. on p. 6)

CHOWDHURY, N. R.; MORĂRESCU, I.; MARTIN, S.; SRIKANT, S. Continuous opinions and discrete actions in social networks: A multi-agent system approach. In: **IEEE 55th Conference on Decision and Control**. Las Vegas, NV, USA: IEEE, 2016. p. 1739–1744. (cit. on p. 4)

CLIFFORD, P.; SUDBURY, A. A model for spatial conflict. **Biometrika**, v. 60, n. 3, p. 581–588, Dec 1973. (cit. on pp. 4 and 21)

CORREIA, F. L. d. B.; MORENO, U. F. Decentralized formation tracking for groups of mobile robots with consensus and mpc. **IFAC-PapersOnLine**, v. 48, n. 19, p. 274–279, Aug 2015. 11th IFAC Symposium on Robot Control, Salvador, Brazil. (cit. on p. 3)

COX, J. T. Coalescing random walks and voter model consensus times on the torus in $\mathbb{Z}^d$. **The Annals of Probability**, Institute of Mathematical Statistics, v. 17, n. 4, p. 1333–1366, Oct 1989. (cit. on p. 22)

CSERMELY, P. **Weak Links: The Universal Key to the Stability of Networks and Complex Systems**. Budapest, Hungary: Springer Berlin Heidelberg, 2009. (The Frontiers Collection). (cit. on p. 66)

CYBENKO, G. Dynamic load balancing for distributed memory multiprocessors. **Journal of Parallel and Distributed Computing**, v. 7, n. 2, p. 279–301, Oct 1989. (cit. on p. 4)

DAI, R.; MESBAHI, M. Optimal topology design for dynamic networks. In: **50th IEEE Conference on Decision and Control and European Control Conference**. Orlando, FL, USA: IEEE, 2011. p. 1280–1285. (cit. on p. 7)

DALEY, D. J.; GANI, J. **Epidemic Modelling: An Introduction**. New York, NY, USA: Cambridge University Press, 1999. (Cambridge Studies in Mathematical Biology). (cit. on p. 17)

DE, P.; LIU, Y.; DAS, S. K. An epidemic theoretic framework for vulnerability analysis of broadcast protocols in wireless sensor networks. **IEEE Transactions on Mobile Computing**, v. 8, n. 3, p. 413–425, Mar 2009. (cit. on p. 6)

DEGROOT, M. H. Reaching a consensus. **Journal of the American Statistical Association**, Taylor & Francis, v. 69, n. 345, p. 118–121, Oct 1974. (cit. on p. 3)

DYER, M.; BEUTEL, J.; THIELE, L. S-xtc: A signal-strength based topology control algorithm for sensor networks. In: **Eighth International Symposium on**

Autonomous Decentralized Systems (ISADS'07). Sedona, AZ, USA: IEEE, 2007. p. 508–518. (cit. on p. 4)

EUGSTER, P. T.; GUERRAOUI, R.; KERMARREC, A.-M.; MASSOULIÉ, L. Epidemic information dissemination in distributed systems. **Computer**, IEEE, v. 37, n. 5, p. 60–67, 2004. (cit. on pp. 6 and 17)

FANTI, M. P.; MANGINI, A. M.; UKOVICH, W. A quantized consensus algorithm for distributed task assignment. In: **51st IEEE Conference on Decision and Control**. Maui, HI, USA: IEEE, 2012. p. 2040–2045. (cit. on p. 4)

FAX, J. A.; MURRAY, R. M. Information flow and cooperative control of vehicle formations. **IEEE Transactions on Automatic Control**, v. 49, n. 9, p. 1465–1476, September 2004. (cit. on p. 3)

FENG, Z.; SUN, C.; HU, G. Robust connectivity preserving rendezvous of multi-robot systems under unknown dynamics and disturbances. In: **IEEE 54th Annual Conference on Decision and Control**. Osaka, Japan: IEEE, 2015. p. 4266–4271. (cit. on p. 5)

FIACCHINI, M.; MORARESCU, I.-C. Convex conditions on decentralized control for graph topology preservation. **IEEE Transactions on Automatic Control**, v. 59, n. 6, p. 1640–1645, Jun 2014. (cit. on p. 5)

FIEDLER, M. Algebraic connectivity of graphs. **Czechoslovak Mathematical Journal**, v. 23, n. 2, p. 298–305, 1973. (cit. on pp. 12 and 15)

FOTOUHI, B.; RABBAT, M. G. Voter model with arbitrary degree dependence: Clout, confidence and irreversibility. **The European Physical Journal B**, Springer, v. 87, n. 3, p. 1–16, Mar 2014. (cit. on p. 22)

FRANCESCHELLI, M.; ROSA, D.; SEATZU, C.; BULLO, F. Gossip algorithms for heterogeneous multi-vehicle routing problems. **Nonlinear Analysis: Hybrid Systems**, v. 10, p. 156–174, Nov 2013. Special Issue related to IFAC Conference on Analysis and Design of Hybrid Systems. (cit. on p. 4)

FREDERICKSON, G. N.; JA'JA', J. On the relationship between the biconnectivity augmentation and travelling salesman problems. **Theoretical Computer Science**, v. 19, n. 2, p. 189–201, Aug 1982. (cit. on pp. 25 and 26)

GANESH, A.; MASSOULIE, L.; TOWSLEY, D. The effect of network topology on the spread of epidemics. In: **Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies**. Miami, FL, USA: IEEE, 2005. p. 1455–1466. (cit. on p. 17)

GASPARRI, A.; SABATTINI, L.; ULIVI, G. Bounded control law for global connectivity maintenance in cooperative multirobot systems. **IEEE Transactions on Robotics**, v. 33, n. 3, p. 700–717, Jun 2017. (cit. on p. 5)

GHEDINI, C.; SECCHI, C.; RIBEIRO, C. H.; SABATTINI, L. Improving robustness in

multi-robot networks. **IFAC-PapersOnLine**, Elsevier, v. 48, n. 19, p. 63–68, Aug 2015. 11th IFAC Symposium on Robot Control, Salvador, Brazil. (cit. on p. 5)

GODSIL, C.; ROYLE, G. **Algebraic graph theory**. 1st. ed. New York: Springer New York, 2001. v. 207. (Graduate Texts in Mathematics, v. 207). (cit. on pp. 12, 13, 14, 15, and 45)

GOFFMAN, W.; NEWILL, V. A. Generalization of epidemic theory: An application to the transmission of ideas. **Nature**, Nature Publishing Group, v. 204, n. 4955, p. 225–228, oct 1964. (cit. on pp. 4, 6, 17, and 43)

GROSS, J. L.; TUCKER, T. W. **Topics in Topological Graph Theory**. New York, NY, USA: Cambridge University Press, 2009. (Encyclopedia of Mathematics and its Applications). (cit. on pp. 11 and 12)

GROSS, T.; BLASIUS, B. Adaptive coevolutionary networks: a review. **Journal of the Royal Society, Interface**, v. 5, n. 20, p. 259–271, mar 2008. (cit. on p. 6)

GROSS, T.; D'LIMA, C. J. D.; BLASIUS, B. Epidemic dynamics on an adaptive network. **Physical Review Letters**, American Physical Society, v. 96, n. 20, p. 208701–4, May 2006. (cit. on p. 6)

GUSRIALDI, A.; QU, Z.; HIRCHE, S. Distributed link removal using local estimation of network topology. **IEEE Transactions on Network Science and Engineering**, v. 6, n. 3, p. 280–292, Sep 2019. (cit. on p. 6)

HAGHIGHI, M. S.; WEN, S.; XIANG, Y.; QUINN, B.; ZHOU, W. On the race of worms and patches: Modeling the spread of information in wireless sensor networks. **IEEE Transactions on Information Forensics and Security**, v. 11, n. 12, p. 2854–2865, Aug 2016. (cit. on p. 6)

HOLLEY, R. A.; LIGGETT, T. M. Ergodic theorems for weakly interacting infinite systems and the voter model. **The Annals of Probability**, v. 3, n. 4, p. 643–663, 1975. (cit. on pp. 4 and 21)

HUNG, P. D.; VINH, T. Q.; NGO, T. D. Hierarchical distributed control for global network integrity preservation in multirobot systems. **IEEE Transactions on Cybernetics**, v. 50, n. 3, p. 1278–1291, Mar 2020. (cit. on p. 5)

JADBABAIE, A.; LIN, J.; MORSE, A. S. Coordination of groups of mobile autonomous agents using nearest neighbor rules. **IEEE Transactions on Automatic Control**, IEEE, v. 48, n. 6, p. 988–1001, jun 2003. (cit. on p. 3)

JIA, P.; FRIEDKIN, N. E.; BULLO, F. Opinion dynamics and social power evolution over reducible influence networks. **SIAM Journal on Control and Optimization**, v. 55, n. 2, p. 1280–1301, Apr 2017. (cit. on p. 6)

KAR, S.; MOURA, J. M. F. Sensor networks with random links: Topology design for distributed consensus. **IEEE Transactions on Signal Processing**, v. 56, n. 7, p. 3315–3326, Jul 2008. (cit. on p. 7)

KASHYAP, A.; BAŞAR, T.; SRIKANT, R. Quantized consensus. **Automatica**, v. 43, n. 7, p. 1192–1203, Jul 2007. (cit. on p. 4)

KEMPE, D.; DOBRA, A.; GEHRKE, J. Gossip-based computation of aggregate information. In: **44th Annual IEEE Symposium on Foundations of Computer Science**. Cambridge, MA, USA: IEEE, 2003. p. 482–491. (cit. on p. 3)

KEPHART, J. O.; WHITE, S. R. Directed-graph epidemiological models of computer viruses. In: **Proceedings of the IEEE Computer Society Symposium on Research in Security and Privacy**. Oakland, CA, USA: IEEE, 1991. p. 343–359. (cit. on pp. 4, 6, and 17)

KHATERI, K.; POURGHOLI, M.; MONTAZERI, M.; SABATTINI, L. A connectivity preserving node permutation local method in limited range robotic networks. **Robotics and Autonomous Systems**, v. 129, n. 103540, p. 1–8, Jul 2020. (cit. on p. 5)

KHELIL, A.; BECKER, C.; TIAN, J.; ROTHERMEL, K. An epidemic model for information diffusion in manets. In: **Proceedings of the 5th ACM International Workshop on Modeling Analysis and Simulation of Wireless and Mobile Systems**. Atlanta, Georgia, USA: ACM Press, 2002. p. 54–60. (cit. on p. 6)

KIBANGOU, A. Y. Graph laplacian based matrix design for finite-time distributed average consensus. In: **American Control Conference**. Montreal, QC, Canada: IEEE, 2012. p. 1901–1906. (cit. on p. 7)

KIM, Y.; MESBAHI, M. On maximizing the second smallest eigenvalue of a state-dependent graph laplacian. **IEEE Transactions on Automatic Control**, v. 51, n. 1, p. 116–120, Jan 2006. (cit. on p. 5)

KIRKLAND, S.; ROCHA, I.; TREVISAN, V. Algebraic connectivity of k-connected graphs. **Czechoslovak Mathematical Journal**, v. 65, n. 1, p. 219–236, Mar 2015. (cit. on p. 12)

LAGORIO, C.; DICKISON, M.; VAZQUEZ, F.; BRAUNSTEIN, L. A.; MACRI, P. A.; MIGUELES, M. V.; HAVLIN, S.; STANLEY, H. E. Quarantine-generated phase transition in epidemic spreading. **Physical Review E**, American Physical Society, v. 83, n. 2, p. 026102–6, Feb 2011. (cit. on p. 7)

LASHHAB, F.; MOORE, K.; VINCENT, T.; MENG, D.; KUWAIRI, K. Robust h_∞ controller design for dynamic consensus networks. **International Journal of Control**, v. 91, n. 8, p. 1906–1919, Jun 2017. (cit. on p. 3)

LI, K.; GONG, R.; WU, S.; HU, C.; WANG, Y. Decentralized robust connectivity control in flocking of multi-robot systems. **IEEE Access**, v. 8, p. 105250–105262, Jun 2020. (cit. on p. 5)

LI, N.; HOU, J. C. Flss: A fault-tolerant topology control algorithm for wireless networks. In: **Proceedings of the 10th Annual International Conference on Mobile Computing and Networking**. New York, NY, USA: ACM Press, 2004. p. 275–286. (cit. on p. 4)

LI, N.; HOU, J. C.; SHA, L. Design and analysis of an mst-based topology control algorithm. In: **IEEE Computer and Communications Societies**. San Francisco, CA, USA: IEEE, 2003. v. 3, p. 1702–1712. (cit. on p. 4)

LIGGETT, T. M. **Interacting Particle Systems**. Los Angeles, CA, USA: Springer, Berlin, Heidelberg, 2005. (Classics in Mathematics). (cit. on p. 22)

LIU, H.; CHU, X.; YIU-LEUNG; DU, R. Simple movement control algorithm for bi-connectivity in robotic sensor networks. **IEEE Journal on Selected Areas in Communications**, v. 28, n. 7, p. 994–1005, Sep 2010. (cit. on p. 5)

LIU, J.; PARÉ, P. E.; NEDIĆ, A.; TANG, C. Y.; BECK, C. L.; BAŞAR, T. On the analysis of a continuous-time bi-virus model. In: **IEEE 55th Conference on Decision and Control**. Las Vegas, NV, USA: IEEE, 2016. p. 290–295. (cit. on p. 43)

LIU, S.; CHEN, P.-Y.; HERO, A. O. Accelerated distributed dual averaging over evolving networks of growing connectivity. **IEEE Transactions on Signal Processing**, v. 66, n. 7, p. 1845–1859, Apr 2018. (cit. on p. 7)

LOBO, M. S.; VANDENBERGHE, L.; BOYD, S.; LEBRET, H. Applications of second-order cone programming. **Linear Algebra and its Applications**, North-Holland, v. 284, n. 1-3, p. 193–228, nov 1998. (cit. on pp. 55 and 56)

LU, W.; CHEN, W.; LAKSHMANAN, L. V. S. From competition to complementarity: Comparative influence diffusion and maximization. **Proceedings of the VLDB Endowment**, ACM, v. 9, n. 2, p. 60–71, Oct 2015. (cit. on p. 6)

LUO, W.; SYCARA, K. Minimum k-connectivity maintenance for robust multi-robot systems. In: **2019 IEEE/RSJ International Conference on Intelligent Robots and Systems**. Macau, China: IEEE, 2019. p. 7370–7377. (cit. on p. 5)

LUO, W.; YI, S.; SYCARA, K. **Behavior Mixing with Minimum Global and Subgroup Connectivity Maintenance for Large-Scale Multi-Robot Systems**. 2019. ArXiv. Disponível em: <https://arxiv.org/abs/1910.01693>. (cit. on p. 5)

MAGGS, M. K.; O'KEEFE, S. G.; THIEL, D. V. Consensus clock synchronization for wireless sensor networks. **IEEE Sensors Journal**, v. 12, n. 6, p. 2269–2277, June 2012. (cit. on p. 3)

MAJCHERCZYK, N.; JAYABALAN, A.; BELTRAME, G.; PINCIROLI, C. Decentralized connectivity-preserving deployment of large-scale robot swarms. In: **2018 IEEE/RSJ International Conference on Intelligent Robots and Systems**. Madrid, Spain: IEEE, 2018. p. 4295–4302. (cit. on p. 4)

MARCEAU, V.; NOËL, P.-A.; HÉBERT-DUFRESNE, L.; ALLARD, A.; DUBÉ, L. J. Adaptive networks: Coevolution of disease and topology. **Physical Review E**, American Physical Society, v. 82, n. 3, p. 036116–10, Sep 2010. (cit. on p. 6)

MARTINS, A. C. R. Continuous opinions and discrete actions in opinion dynamics problems. **International Journal of Modern Physics C**, World Scientific Publishing

Company, v. 19, n. 4, p. 617–624, apr 2008. (cit. on p. 4)

MERRIS, R. Laplacian matrices of graphs: a survey. **Linear Algebra and its Applications**, North-Holland, v. 197-198, p. 143–176, Jan 1994. (cit. on p. 14)

MESBAHI, M.; EGERSTEDT, M. **Graph Theoretic Methods in Multiagent Networks**. 1st. ed. Princeton, New Jersey, USA: Princeton University Press, 2010. (Applied Mathematics). (cit. on p. 13)

MIEGHEM, P. V. **Performance Analysis of Communications Networks and Systems**. New York, NY, USA: Cambridge University Press, 2006. (cit. on p. 46)

MIEGHEM, P. V. The n-intertwined sis epidemic network model. **Computing**, v. 93, n. 2, p. 147–169, Dec 2011. (cit. on pp. 17 and 44)

MIEGHEM, P. V.; OMIC, J.; KOOIJ, R. Virus spread in networks. **IEEE/ACM Transactions on Networking**, v. 17, n. 1, p. 1–14, Feb 2009. (cit. on pp. 17, 43, and 44)

MILLER, C. E.; TUCKER, A. W.; ZEMLIN, R. A. Integer programming formulation of traveling salesman problems. **Journal of the ACM**, ACM, v. 7, n. 4, p. 326–329, Oct 1960. (cit. on p. 25)

MIROLLO, R. E.; STROGATZ, S. H. Synchronization of pulse-coupled biological oscillators. **SIAM Journal on Applied Mathematics**, v. 50, n. 6, p. 1645–1662, Jul 1990. (cit. on p. 3)

MIRTABATABAEI, A.; BULLO, F. Opinion dynamics in heterogeneous networks: Convergence conjectures and theorems. **SIAM Journal on Control and Optimization**, v. 50, n. 5, p. 2763–2785, Sep 2012. (cit. on p. 6)

MOALLEMI, C. C.; ROY, B. V. Consensus propagation. **IEEE Transactions on Information Theory**, v. 52, n. 11, p. 4753–4766, Nov 2006. (cit. on p. 3)

MOHAR, B. The laplacian spectrum of graphs. In: **Graph Theory, Combinatorics, and Applications**. New York, NY, USA: Wiley, 1991. p. 871–898. (cit. on p. 15)

MURRAY, R. M. Recent research in cooperative control of multivehicle systems. **Journal of Dynamic Systems, Measurement, and Control**, American Society of Mechanical Engineers, v. 129, n. 5, p. 571–583, May 2007. (cit. on p. 13)

NEWMAN, M. Measures and metrics. In: **Networks: An Introduction**. New York, NY, USA: Oxford University Press, 2010. ch. 7, p. 168–234. (cit. on pp. 15 and 16)

NEWMAN, M. E. J. Mathematics of networks. In: **Networks: An Introduction**. New York, NY, USA: Oxford University Press, 2010. ch. 6, p. 109–167. (cit. on p. 14)

NOORAZAR, H. Recent advances in opinion propagation dynamics: A 2020 survey. **The European Physical Journal Plus**, v. 135, n. 521, p. 1–20, Jun 2020. (cit. on p. 4)

OH, K.-K.; MOORE, K. L.; AHN, H.-S. Disturbance attenuation in consensus network

of identical linear systems: an h_∞ approach. **IEEE Transactions on Automatic Control**, v. 59, n. 8, p. 2164–2169, Aug 2014. (cit. on p. 3)

OJHA, R. P.; SRIVASTAVA, P. K.; SANYAL, G. Improving wireless sensor networks performance through epidemic model. **International Journal of Electronics**, Taylor & Francis, v. 106, n. 6, p. 862–879, Feb 2019. (cit. on p. 6)

OLFATI-SABER, R. Distributed kalman filter with embedded consensus filters. In: **Proceedings of the 44th IEEE Conference on Decision and Control**. Seville, Spain: IEEE, 2005. p. 8179–8184. (cit. on p. 3)

OLFATI-SABER, R. Ultrafast consensus in small-world networks. In: **Proceedings of the American Control Conference**. Portland, OR, USA: IEEE, 2005. p. 2371–2378. (cit. on p. 7)

OLFATI-SABER, R.; FAX, J. A.; MURRAY, R. M. Consensus and cooperation in networked multi-agent systems. **Proceedings of the IEEE**, v. 95, n. 1, p. 215–233, Jan 2007. (cit. on pp. 3, 13, 19, and 20)

OLFATI-SABER, R.; MURRAY, R. M. Consensus protocols for networks of dynamic agents. **Proceedings of the American Control Conference**, v. 2, p. 951–956, 2003. (cit. on pp. 13 and 18)

OLFATI-SABER, R.; MURRAY, R. M. Consensus problems in networks of agents with switching topology and time-delays. **IEEE Transactions on Automatic Control**, v. 49, n. 9, p. 1520–1533, 2004. (cit. on p. 3)

OLFATI-SABER, R.; SHAMMA, J. S. Consensus filters for sensor networks and distributed sensor fusion. In: **Proceedings of the 44th IEEE Conference on Decision and Control**. Seville, Spain: IEEE, 2005. p. 6698–6703. (cit. on p. 3)

ORDOÑEZ, B.; MORENO, U. F.; CERQUEIRA, J.; ALMEIDA, L. Generation of trajectories using predictive control for tracking consensus with sensing. **Procedia Computer Science**, v. 10, p. 1094–1099, 2012. (cit. on pp. 3 and 27)

OROSZ, G.; MOEHLIS, J.; MURRAY, R. M. Controlling biological networks by time-delayed signals. **Philosophical Transactions of the Royal Society of A**, v. 368, n. 1911, p. 439–454, Jan 2010. (cit. on p. 3)

PAPOULIS, A.; PILLAI, S. U. **Probability, Random Variables, and Stochastic Processes**. 4th. ed. New York, NY, USA: McGraw-Hill Higher Education, 2002. (cit. on p. 44)

PASTOR-SATORRAS, R.; VESPIGNANI, A. Epidemic spreading in scale-free networks. **Physical Review Letters**, American Physical Society, v. 86, n. 14, p. 3200–3203, Apr 2001. (cit. on p. 6)

PENROSE, M. **Random Geometric Graphs**. New York, NY, USA: Oxford University Press, 2003. (Oxford Studies in Probability). (cit. on p. 58)

PRASETYO, J.; MASI, G. D.; FERRANTE, E. Collective decision making in dynamic

environments. **Swarm Intelligence**, v. 13, p. 217–243, Jun 2019. (cit. on p. 4)

PRECIADO, V. M.; VERGHESE, G. C. Synchronization in generalized erdös-rényi networks of nonlinear oscillators. In: **44th IEEE Conference on Decision and Control, and the European Control Conference**. Seville, Spain: IEEE, 2005. p. 4628–4633. (cit. on p. 3)

RAFIEE, M.; BAYEN, A. M. Optimal network topology design in multi-agent systems for efficient average consensus. In: **49th IEEE Conference on Decision and Control**. Atlanta, GA, USA: IEEE, 2010. p. 3877–3883. (cit. on pp. 7 and 56)

RAMALHO, G. M.; CARVALHO, S. R.; FINARDI, E. C.; MORENO, U. F. Trajectory optimization using sequential convex programming with collision avoidance. **Journal of Control, Automation and Electrical Systems**, Springer US, v. 29, n. 3, p. 318–327, Jun 2018. (cit. on p. 71)

REDNER, S. Reality-inspired voter models: A mini-review. **Comptes Rendus Physique**, Elsevier Masson SAS, v. 20, n. 4, p. 275–292, May 2019. (cit. on p. 22)

REN, W.; BEARD, R. W. **Distributed Consensus in Multi-vehicle Cooperative Control**. 1st. ed. London: Springer London, 2008. (Communications and Control Engineering). (cit. on pp. 3 and 13)

RISAU-GUSMAN, S.; ZANETTE, D. H. Contact switching as a control strategy for epidemic outbreaks. **Journal of Theoretical Biology**, v. 257, n. 1, p. 52–60, Mar 2009. (cit. on p. 7)

ROY, S.; HERLUGSON, K.; SABERI, A. A control-theoretic approach to distributed discrete-valued decision-making in networks of sensing agents. **IEEE Transactions on Mobile Computing**, v. 5, n. 8, p. 945–957, Aug 2006. (cit. on p. 4)

SABATTINI, L.; SECCHI, C.; CHOPRA, N.; GASPARRI, A. Distributed control of multirobot systems with global connectivity maintenance. **IEEE Transactions on Robotics**, v. 29, n. 5, p. 1326–1332, Oct 2013. (cit. on p. 5)

SALEM, F.; TCHILIAN, R.; CARVALHO, S.; MORENO, U. Opinion dynamics over a finite set in cooperative multi-robot systems: An asynchronous gossip-based consensus approach. In: **XXIII Congresso Brasileiro de Automática**. Porto Alegre, RS, Brazil: Galoá, 2020. (cit. on p. 71)

SALEM, F. A.; MORENO, U. F.; CASTELAN, E. B. Information distributed kalman filter applied to rendezvous problems in cooperative robotic teams. **IFAC-PapersOnLine**, v. 51, n. 25, p. 190–195, Sep 2018. 9th IFAC Symposium on Robust Control Design, Florianópolis, Brazil. (cit. on p. 3)

SALEM, F. A.; TCHILIAN, R. da S.; MORENO, U. Evolution of discrete opinions on human-swarm interaction in influence networks. In: **14º Simpósio Brasileiro de Automação Inteligente**. Ouro Preto, MG, Brazil: Galoá, 2019. v. 1, p. 2344–2349. (cit. on p. 3)

SCHENATO, L.; FIORENTIN, F. Average timesynch: A consensus-based protocol for clock synchronization in wireless sensor networks. **Automatica**, v. 47, n. 9, p. 1878–1886, July 2011. (cit. on p. 3)

SCHURESKO, M.; CORTÉS, J. Distributed motion constraints for algebraic connectivity of robotic networks. **Journal of Intelligent and Robotic Systems**, v. 56, n. 1, p. 99–126, Sep 2009. (cit. on p. 5)

SHAH, D. Gossip algorithms. **Foundations and Trends in Networking**, v. 3, n. 1, p. 1–125, Jun 2007. (cit. on p. 17)

SHAN, Q.; TENG, F.; LI, T.; CHEN, C. L. P. Containment control of multi-agent systems with nonvanishing disturbance via topology reconfiguration. **Science China Information Sciences**, v. 64, n. 179203, p. 1–3, May 2020. (cit. on p. 7)

SHAN, Q.; YAN, J.; LI, T.; CHEN, C. L. P. Containment control of multi-agent systems with general noise based on hierarchical topology reconfiguration. **IEEE Access**, v. 7, p. 56826–56838, Apr 2019. (cit. on p. 7)

SHARKEY, K. J. Deterministic epidemiological models at the individual level. **Journal of Mathematical Biology**, Springer-Verlag, v. 57, n. 3, p. 311–331, Feb 2008. (cit. on p. 17)

SHAW, L. B.; SCHWARTZ, I. B. Fluctuating epidemics on adaptive networks. **Physical Review E**, American Physical Society, v. 77, n. 6, p. 066101–10, Jun 2008. (cit. on p. 6)

SOOD, V.; REDNER, S. Voter model on heterogeneous graphs. **Physical Review Letters**, American Physical Society, v. 94, n. 17, p. 178701–4, May 2005. (cit. on p. 22)

STROGATZ, S. H. From kuramoto to crawford: Exploring the onset of synchronization in populations of coupled oscillators. **Physica D: Nonlinear Phenomena**, v. 143, n. 1, p. 1–20, Sep 2000. (cit. on p. 3)

SVANBERG, K. The method of moving asymptotes - a new method for structural optimization. **International Journal for Numerical Methods In Engineering**, v. 24, n. 2, p. 359–373, Feb 1987. (cit. on p. 29)

SYDNEY, A.; SCOGLIO, C.; GRUENBACHER, D. Optimizing algebraic connectivity by edge rewiring. **Applied Mathematics and Computation**, v. 219, n. 10, p. 5465–5479, Jan 2013. (cit. on p. 5)

TALEBI, S. P.; WERNER, S. Distributed kalman filtering and control through embedded average consensus information fusion. **IEEE Transactions on Automatic Control**, v. 64, n. 10, p. 4396–4403, Oct 2019. (cit. on p. 3)

TARDÓS, J.; ARAGUES, R.; SAGÜÉS, C.; RUBIO, C. Simultaneous deployment and tracking multi-robot strategies with connectivity maintenance. **Sensors**, v. 18, n. 3, p. 1–24, Mar 2018. (cit. on p. 5)

TRAJANOVSKI, S.; GUO, D.; MIEGHEM, P. V. From epidemics to information propagation: Striking differences in structurally similar adaptive network models.

Physical Review E, v. 92, n. 3, p. 030801–5, Sep 2015. (cit. on p. 6)

VALENTINI, G. **Achieving Consensus in Robot Swarms: Design and Analysis of Strategies for the best-of-n Problem**. Tempe, AZ, USA: Springer, 2017. (Studies in Computational Intelligence). (cit. on p. 57)

VALENTINI, G.; FERRANTE, E.; DORIGO, M. The best-of-n problem in robot swarms: Formalization, state of the art, and novel perspectives. **Frontiers in Robotics and AI**, v. 4, n. 9, p. 1–18, Mar 2017. (cit. on p. 4)

VALENTINI, G.; HAMANN, H.; DORIGO, M. Self-organized collective decision making: the weighted voter model. In: **Proceedings of the 2014 International Conference on Autonomous Agents and Multi-Agent Systems**. Paris, France: International Foundation for Autonomous Agents and Multiagent Systems, 2014. p. 45–52. (cit. on pp. 4 and 22)

WANG, Y.; CHAKRABARTI, D.; WANG, C.; FALOUTSOS, C. Epidemic spreading in real networks: an eigenvalue viewpoint. In: **Proceedings of the 22nd International Symposium on Reliable Distributed Systems**. Florence, Italy: IEEE, 2003. (cit. on p. 17)

XIAO, L.; BOYD, S. Fast linear iterations for distributed averaging. **Systems & Control Letters**, v. 53, n. 1, p. 65–78, Sep 2004. (cit. on pp. 7 and 19)

XIAO, L.; BOYD, S.; LALL, S. A scheme for robust distributed sensor fusion based on average consensus. In: **Fourth International Symposium on Information Processing in Sensor Networks**. Boise, ID, USA: IEEE, 2005. p. 63–70. (cit. on p. 3)

XIAO, Y.-P.; LI, S.-Y.; LIU, Y.-B. An information diffusion dynamic model based on social influence and mean-field theory. **Acta Physica Sinica**, v. 66, n. 3, p. 030501–13, Feb 2017. (cit. on p. 6)

XUE, D.; HIRCHE, S. Distributed topology manipulation to control epidemic spreading over networks. **IEEE Transactions on Signal Processing**, v. 67, n. 5, p. 1163–1174, Mar 2019. (cit. on p. 6)

XUE, D.; HIRCHE, S. Finite-time distributed topology design for optimal network resilience. **IET Control Theory Applications**, v. 13, n. 17, p. 2792–2799, Nov 2019. (cit. on p. 7)

YANG, L.-X.; LI, P.; YANG, X.; WU, Y.; TANG, Y. Y. On the competition of two conflicting messages. **Nonlinear Dynamics**, Springer Netherlands, v. 91, n. 3, p. 1853–1869, feb 2018. (cit. on p. 43)

YANG, P.; FREEMAN, R. A.; GORDON, G. J.; LYNCH, K. M.; SRINIVASA, S. S.; SUKTHANKAR, R. Decentralized estimation and control of graph connectivity for mobile sensor networks. **Automatica**, v. 46, n. 2, p. 390–396, 2010. (cit. on p. 5)

YANG, X.; FAN, X.; LI, G. Decentralized cooperative control with connectivity maintenance for multiagent systems. **International Robotics & Automation**

Journal, v. 5, n. 1, p. 1–9, Jan 2019. (cit. on p. 5)

YILDIZ, E.; ACEMOGLU, D.; OZDAGLAR, A. E.; SABERI, A.; SCAGLIONE, A. Discrete opinion dynamics with stubborn agents. **SSRN Electronic Journal**, p. 1–39, Jan 2011. (cit. on p. 22)

YILDIZ, M. E.; PAGLIARI, R.; OZDAGLAR, A.; SCAGLIONE, A. Voting models in random networks. In: **Information Theory and Applications Workshop**. San Diego, CA, USA: IEEE, 2010. p. 1–7. (cit. on pp. 4, 21, and 22)

ZAREH, M.; SABATTINI, L.; SECCHI, C. Decentralized biconnectivity conditions in multi-robot systems. In: **IEEE 55th Conference on Decision and Control**. Las Vegas, NV, USA: IEEE, 2016. (cit. on p. 15)

ZAREH, M.; SABATTINI, L.; SECCHI, C. Enforcing biconnectivity in multi-robot systems. In: **IEEE 55th Conference on Decision and Control**. Las Vegas, NV, USA: IEEE, 2016. p. 1800–1805. (cit. on p. 5)

ZAVLANOS, M. M.; PAPPAS, G. J. Controlling connectivity of dynamic graphs. In: **Proceedings of the 44th IEEE Conference on Decision and Control**. Seville, Spain: IEEE, 2005. p. 6388–6393. (cit. on p. 5)

ZAVLANOS, M. M.; PRECIADO, V. M.; JADBABAIE, A. Spectral control of mobile robot networks. In: **Proceedings of the 2011 American Control Conference**. San Francisco, CA, USA: IEEE, 2011. p. 3245–3250. (cit. on p. 5)

ZHAO, L.; JIA, Y.; YU, J.; DU, J. h_∞ sliding mode based scaled consensus control for linear multi-agent systems with disturbances. **Applied Mathematics and Computation**, v. 292, n. 1, p. 375–389, Jan 2017. (cit. on p. 3)

ZILLOBER, C.; SCHITTKOWSKI, K.; MORITZEN, K. Very large scale optimization by sequential convex programming. **Optimization Methods and Software**, v. 19, n. 1, p. 103–120, May 2004. (cit. on p. 29)