

Jinhu Lü
Xinghuo Yu
Guanrong Chen
Wenwu Yu *Editors*

Complex Systems and Networks

Dynamics, Controls and Applications



Springer

Springer Complexity

Springer Complexity is an interdisciplinary program publishing the best research and academic-level teaching on both fundamental and applied aspects of complex systems—cutting across all traditional disciplines of the natural and life sciences, engineering, economics, medicine, neuroscience, social and computer science.

Complex Systems are systems that comprise many interacting parts with the ability to generate a new quality of macroscopic collective behavior the manifestations of which are the spontaneous formation of distinctive temporal, spatial or functional structures. Models of such systems can be successfully mapped onto quite diverse “real-life” situations like the climate, the coherent emission of light from lasers, chemical reaction-diffusion systems, biological cellular networks, the dynamics of stock markets and of the Internet, earthquake statistics and prediction, freeway traffic, the human brain, or the formation of opinions in social systems, to name just some of the popular applications.

Although their scope and methodologies overlap somewhat, one can distinguish the following main concepts and tools: self-organization, nonlinear dynamics, synergetics, turbulence, dynamical systems, catastrophes, instabilities, stochastic processes, chaos, graphs and networks, cellular automata, adaptive systems, genetic algorithms and computational intelligence.

The three major book publication platforms of the Springer Complexity program are the monograph series “Understanding Complex Systems” focusing on the various applications of complexity, the “Springer Series in Synergetics”, which is devoted to the quantitative theoretical and methodological foundations, and the “Springer Briefs in Complexity” which are concise and topical working reports, case studies, surveys, essays and lecture notes of relevance to the field. In addition to the books in these two core series, the program also incorporates individual titles ranging from textbooks to major reference works.

Editorial and Programme Advisory Board

Henry Abarbanel, Institute for Nonlinear Science, University of California, San Diego, USA

Dan Braha, New England Complex Systems Institute and University of Massachusetts Dartmouth, USA

Péter Érdi, Center for Complex Systems Studies, Kalamazoo College, USA and Hungarian Academy of Sciences, Budapest, Hungary

Karl Friston, Institute of Cognitive Neuroscience, University College London, London, UK

Hermann Haken, Center of Synergetics, University of Stuttgart, Stuttgart, Germany

Viktor Jirsa, Centre National de la Recherche Scientifique (CNRS), Université de la Méditerranée, Marseille, France

Janusz Kacprzyk, System Research, Polish Academy of Sciences, Warsaw, Poland

Kunihiko Kaneko, Research Center for Complex Systems Biology, The University of Tokyo, Tokyo, Japan

Scott Kelso, Center for Complex Systems and Brain Sciences, Florida Atlantic University, Boca Raton, USA

Markus Kirkilionis, Mathematics Institute and Centre for Complex Systems, University of Warwick, Coventry, UK

Jürgen Kurths, Nonlinear Dynamics Group, University of Potsdam, Potsdam, Germany

Andrzej Nowak, Department of Psychology, Warsaw University, Poland

Hassan Qudrat-Ullah, School of Administrative Studies, York University, Toronto, ON, Canada

Linda Reichl, Center for Complex Quantum Systems, University of Texas, Austin, USA

Peter Schuster, Theoretical Chemistry and Structural Biology, University of Vienna, Vienna, Austria

Frank Schweitzer, System Design, ETH Zürich, Zürich, Switzerland

Didier Sornette, Entrepreneurial Risk, ETH Zürich, Zürich, Switzerland

Stefan Thurner, Section for Science of Complex Systems, Medical University of Vienna, Vienna, Austria

Understanding Complex Systems

Founding Editor: S. Kelso

Future scientific and technological developments in many fields will necessarily depend upon coming to grips with complex systems. Such systems are complex in both their composition – typically many different kinds of components interacting simultaneously and nonlinearly with each other and their environments on multiple levels – and in the rich diversity of behavior of which they are capable.

The Springer Series in Understanding Complex Systems series (UCS) promotes new strategies and paradigms for understanding and realizing applications of complex systems research in a wide variety of fields and endeavors. UCS is explicitly transdisciplinary. It has three main goals: First, to elaborate the concepts, methods and tools of complex systems at all levels of description and in all scientific fields, especially newly emerging areas within the life, social, behavioral, economic, neuro- and cognitive sciences (and derivatives thereof); second, to encourage novel applications of these ideas in various fields of engineering and computation such as robotics, nano-technology, and informatics; third, to provide a single forum within which commonalities and differences in the workings of complex systems may be discerned, hence leading to deeper insight and understanding.

UCS will publish monographs, lecture notes, and selected edited contributions aimed at communicating new findings to a large multidisciplinary audience.

More information about this series at <http://www.springer.com/series/5394>

Jinhu Lü · Xinghuo Yu · Guanrong Chen
Wenwu Yu
Editors

Complex Systems and Networks

Dynamics, Controls and Applications



Springer

Editors

Jinhu Lü
Institute of Systems Science
Academy of Mathematics and Systems
Science, Chinese Academy of Sciences
Beijing
China

Xinghuo Yu
School of Electrical and Computer
Engineering
RMIT University
Melbourne
Australia

Guanrong Chen
Department of Electronic Engineering
City University of Hong Kong
Hong Kong
China

Wenwu Yu
Department of Mathematics
Southeast University
Nanjing
China

ISSN 1860-0832
Understanding Complex Systems
ISBN 978-3-662-47823-3
DOI 10.1007/978-3-662-47824-0

ISSN 1860-0840 (electronic)
ISBN 978-3-662-47824-0 (eBook)

Library of Congress Control Number: 2015945120

Springer Heidelberg New York Dordrecht London
© Springer-Verlag Berlin Heidelberg 2016

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made.

Printed on acid-free paper

Springer-Verlag GmbH Berlin Heidelberg is part of Springer Science+Business Media
(www.springer.com)

Preface

Nowadays, networks exist everywhere. In the recent decade, complex networks have been widely investigated partly due to their wide applications in biological neural networks, ecosystems, metabolic pathways, the Internet, the WWW, electrical power grids, communication systems, etc., and partly due to their broad scientific progress in physics, mathematics, engineering, biology, etc. The key character for a complex network is that it can represent a large-scale system in nature, human societies, and technology with the nodes representing the individual agents and the edges representing the mutual connections. Thus, the research work on fundamental properties, such as dynamics, controls, and applications of various complex networks has become overwhelming recently.

Actually, complex network studies can be dated back to the eighteenth century when the great mathematician Leonhard Euler studied the well-known Königsburg seven-bridge problem. Then, in the early 1960s, Erdős and Rényi (ER) proposed a random-graph model, which can be regarded as the modern network theory framework. In order to describe a transition from a regular network to a random network, Watts and Strogatz (WS) rewired the connections on some nodes in a regular network and proposed a small-world network model. Then, Barabási and Albert (BA) proposed a new scale-free network model, in which the degree distribution of the nodes follows a power-law form. Thereafter, complex networks have been widely discussed. In particular, small-world and scale-free complex networks have been extensively investigated worldwide.

The contents of this book are summarized as follows. First, the dynamics of complex networks are studied regarding, for example, the cluster dynamic analysis using kernel spectral methods, community detection algorithms in bipartite networks, epidemiological modeling with demographics and epidemic spreading on multi-layer networks, and resilience of spatial networks leading to the catastrophic cascading failures under various local perturbations. Then, some evolving hyper-network and color-network models are generated by adopting both growth and preferential attachment mechanisms and some new nonlinear chaotic pseudo random number generator, based on tent and logistic maps are also discussed.

Second, the controls of complex networks are investigated. The interesting topics include distributed finite-time cooperative control of multi-agent systems by applying homogeneous-degree and Lyapunov methods, composite finite-time containment control for disturbed second-order multi-agent systems, fractional-order observer design of multi-agent systems, chaos control and anti-control of complex systems via Parrondos game, collective behavior coordination with predictive mechanisms, convergence, consensus and synchronization of complex networks via contraction theory, and structural controllability of temporal complex networks.

Third, the applications of complex networks provide some applicable carriers, which show the importance of theories developed in complex networks. In particular, a general model for studying time evolution of transition networks, deflection routing in complex networks, recommender systems for social networks analysis and mining, strategy selection in networked evolutionary games, integration and methods in computational biology, are discussed in detail.

Recently, studies of the dynamics and controls of complex networks have become more attractive. In particular, some emergent behaviors of complex networks need to be investigated because new applied science and technology require new methods and theories to solve new challenging problems. Thus, an in-depth study with detailed description of dynamics, controls, and applications of complex networks will benefit both theoretical research and applications in the near-future development of related subjects. This book provides some state-of-the-art research results on broad disciplinary sciences in complex networks to meet such demands.

We would like to express our sincere thanks to all the chapter contributors for their great support to our book, without which this book would not have been possible. Special thanks are directed to the founding editor of the Springer Series in Understanding Complex Systems, Scott Kelso, for his encouragement and support to edit this volume. Thanks also go to Dr. Thomas Ditzinger, Holger Schäpe, and Priyadarshini Senthilkumar from Springer for their assistance during the publication of this book. Last but not least, we also would like to thank the financial support from the National Science and Technology Major Project of China under Grant 2014ZX10004001-014, the 973 Project under Grant 2014CB845302, and the National Natural Science Foundation of China under Grant Nos. 11472290, 61322302, and 61104145, Australian Research Council Discovery under Grants Nos. DP130104765 and DP140100544, and Hong Kong Research Grants Council under the GRF Grants CityU 11201414 and 11208515.

Beijing
Melbourne
Hong Kong
Nanjing
May 2015

Jinhu Lü
Xinghuo Yu
Guanrong Chen
Wenwu Yu

Contents

1	Discovering Cluster Dynamics Using Kernel Spectral Methods	1
	Rocco Langone, Raghvendra Mall, Joos Vandewalle and Johan A.K. Suykens	
2	Community Detection in Bipartite Networks: Algorithms and Case studies	25
	Taher Alzahrani and K.J. Horadam	
3	Epidemiological Modeling on Complex Networks	51
	Zhen Jin, Shuping Li, Xiaoguang Zhang, Juping Zhang and Xiao-Long Peng	
4	Resilience of Spatial Networks	79
	Daqing Li	
5	Synchronization and Control of Hyper-Networks and Colored Networks	107
	Xinchu Fu, Zhaoyan Wu and Guanrong Chen	
6	New Nonlinear CPRNG Based on Tent and Logistic Maps	131
	Oleg Garasym, Ina Taralova and René Lozi	
7	Distributed Finite-Time Cooperative Control of Multi-agent Systems	163
	Yu Zhao, Guanghui Wen and Guanrong Chen	
8	Composite Finite-Time Containment Control for Disturbed Second-Order Multi-agent Systems	207
	Xiangyu Wang and Shihua Li	

9 Application of Fractional-Order Calculus in a Class of Multi-agent Systems	229
Wenwu Yu, Guanghui Wen and Yang Li	
10 Chaos Control and Anticontrol of Complex Systems via Parrondo's Game	263
Marius-F. Danca	
11 Collective Behavior Coordination with Predictive Mechanisms	283
Hai-Tao Zhang, Zhaomeng Cheng, Ming-Can Fan and Yue Wu	
12 Convergence, Consensus and Synchronization of Complex Networks via Contraction Theory	313
Mario di Bernardo, Davide Fiore, Giovanni Russo and Francesco Scafuti	
13 Towards Structural Controllability of Temporal Complex Networks	341
Xiang Li, Peng Yao and Yujian Pan	
14 A General Model for Studying Time Evolution of Transition Networks	373
Choujun Zhan, Chi K. Tse and Michael Small	
15 Deflection Routing in Complex Networks	395
Soroush Haeri and Ljiljana Trajkovic	
16 Recommender Systems for Social Networks Analysis and Mining: Precision Versus Diversity	423
Amin Javari, Malihe Izadi and Mahdi Jalili	
17 Strategy Selection in Networked Evolutionary Games: Structural Effect and the Evolution of Cooperation	439
Shaolin Tan and Jinhu Lü	
18 Network Analysis, Integration and Methods in Computational Biology: A Brief Survey on Recent Advances	459
Shihua Zhang	

Chapter 1

Discovering Cluster Dynamics Using Kernel Spectral Methods

Rocco Langone, Raghvendra Mall, Joos Vandewalle
and Johan A.K. Suykens

Abstract Networks represent patterns of interactions between components of complex systems present in nature, science, technology and society. Furthermore, graph theory allows to perform insightful analysis for different kinds of data by representing the instances as nodes of a weighted network, where the weights characterize similarity between the data points. In this chapter we describe a number of algorithms to perform cluster analysis, that is finding groups of similar items (called clusters or communities) and understand their evolution over time. These algorithms are designed in a kernel-based framework: the original data are mapped into an high dimensional feature space; linear models are designed in this space; complex non-linear relationships between the data in the original input space can then be detected. Applications like fault detection in industrial machines, community detection of static and evolving networks, image segmentation, incremental time-series clustering and text clustering are considered.

1.1 Introduction

Graph theory constitutes a powerful tool for data analysis. In fact, by representing the similarity between each pair of data points as a network, complex patterns can be revealed. The most popular class of algorithms based on graph theory is spectral clustering abbreviated as SC [1], which exploits the spectral properties of the so called Laplacian to partition a graph into weakly connected sub-graphs. SC started

R. Langone · R. Mall · J. Vandewalle (✉) · J.A.K. Suykens
Department of Elektrotechniek (ESAT) Stadus, KU Leuven,
Kasteelpark Arenberg 10, B-3001 Leuven, Belgium
e-mail: joos.vandewalle@esat.kuleuven.be

R. Langone
e-mail: rocco.langone@esat.kuleuven.be

R. Mall
e-mail: raghvendra.mall@esat.kuleuven.be

J.A.K. Suykens
e-mail: johan.suykens@esat.kuleuven.be

to become a popular and state-of-the-art algorithm for data clustering after the works of Shi and Malik [2]. They proposed to optimize the Normalized Cut criterion to solve the image segmentation problem. Ng and Jordan [3] described an analysis of the SC algorithm by means of matrix perturbation theory that gives conditions under which a good performance is expected, and the tutorial by Von Luxburg reviewed the main literature related to SC [4]. Although very successful in a variety of applications, SC cannot handle big data without using approximation methods like the Nyström algorithm [5, 6], the power iteration method [7], or techniques based on linear algebra concepts [8–10]. Moreover, the out-of-sample extension is only approximate.

Lately, a spectral clustering algorithm formulated in a kernel framework has been proposed [11]. The method, called kernel spectral clustering (KSC), is based on solving a primal-dual optimization problem typical of Least Squares Support Vector Machines or LS-SVMs [12]. KSC has two main advantages w.r.t. SC: the possibility to perform model selection to detect, for instance, the natural number of clusters which are present in the data, and the out-of-sample extension to unseen test points, by means of a model learned during the training process using a subset of the entire data.

One implicit assumption when using KSC is that the data do not change, i.e. they are so to say static. However, in many real-world scenarios like in industrial process monitoring, scientific experiments, social network activity etc., data are normally time-stamped. In this case, clustering algorithms including a time variable in their formulation are more suitable to discover meaningful patterns and track their evolution over time. Examples of such algorithms are evolutionary (spectral) clustering [13–15] characterized by the temporal smoothness between clusters in successive time-steps, a tensor-based approach proposed in [16], which generalizes the determination of community structure to multi-slice networks defined by coupling multiple adjacency matrices at different times, incremental k-means, which at each time-step uses the previous centroids to find the new cluster centers [17].

In contrast to the aforementioned algorithms which work on the entire data, two generalizations of KSC have been recently proposed to deal with dynamic clustering in a model-based framework. The new techniques are referred as kernel spectral clustering with memory or MKSC [18–20] and incremental kernel spectral clustering abbreviated as IKSC [21]. Concerning the first algorithm, the temporal smoothness between clusters in successive time-steps is incorporated in the primal optimization problem, inspired by the evolutionary clustering approaches. This allows to track the long-term trend of the clusters and to reduce the sensitivity to noisy short-term variations. Moreover, a precise model selection scheme based on smoothed cluster quality measures and the out-of-sample extension to new points make MKSC unique in its kind. The second method, namely IKSC, is particularly suitable to cluster data streams: the model is expressed only by the cluster prototypes in the eigenspace of KSC, and is continuously updated in response to new data. By doing so, complex patterns emerging across time in a non-stationary environment can be revealed.

In the next sections, after recalling the KSC method and some interesting applications where it has been utilized, we will describe the MKSC and IKSC techniques and how they can be employed in different domains to perform dynamic data clustering.

1.2 Notation

x^T	Transpose of a vector x
Ω^T	Transpose of a matrix Ω
Ω_{ij}	i th entry of the matrix Ω
I_N	$N \times N$ Identity matrix
1_N	$N \times 1$ Vector of ones
$\mathcal{D}_{\text{Tr}} = \{x_i\}_{i=1}^{N_{\text{Tr}}}$	Training sample of N_{Tr} data points
$\varphi(\cdot)$	Feature map
\mathcal{F}	Feature space of dimension d_h
$K(x_i, x_j)$	Kernel function evaluated on data points x_i, x_j
$\{\mathcal{A}_p\}_{p=1}^k$	Partitioning composed of k clusters
$\alpha_i \in \mathbb{R}$	i th entry of the dual solution vector $\alpha \in \mathbb{R}^{N_{\text{Tr}}}$
D	$N \times N$ graph degree matrix
$\mathcal{G} = (\mathcal{V}, \mathcal{E})$	Set of N vertices $\mathcal{V} = \{v_i\}_{i=1}^N$ and m edges \mathcal{E} of a graph
$\mathcal{S} = \{(\mathcal{V}_t, \mathcal{E}_t)\}_{t=1}^T$	Sequence of graphs over time T
$ \cdot $	Cardinality of a set

1.3 Static Clustering

1.3.1 The KSC Model

Given a training data set $\mathcal{D}_{\text{Tr}} = \{x_i\}_{i=1}^{N_{\text{Tr}}}$, the multi-cluster KSC model [11] is expressed by $k - 1$ binary problems, where k indicates the number of clusters:

$$\begin{aligned} \min_{w^{(l)}, e^{(l)}, b_l} \quad & \frac{1}{2} \sum_{l=1}^{k-1} w^{(l)T} w^{(l)} - \frac{1}{2N_{\text{Tr}}} \sum_{l=1}^{k-1} \gamma_l e^{(l)T} V e^{(l)} \\ \text{subject to} \quad & e^{(l)} = \Phi w^{(l)} + b_l 1_{N_{\text{Tr}}}, l = 1, \dots, k-1. \end{aligned} \quad (1.1)$$

The $e^{(l)} = [e_1^{(l)}, \dots, e_i^{(l)}, \dots, e_{N_{\text{Tr}}}^{(l)}]^T$ are the projections of all the training data points mapped in the feature space along the direction $w^{(l)}$. For a given point x_i , the primal clustering model is expressed by:

$$e_i^{(l)} = w^{(l)T} \varphi(x_i) + b_l. \quad (1.2)$$

The optimization problem (1.1) means the maximization of the weighted variances $C_l = e^{(l)T} V e^{(l)}$ regularized by the minimization of the squared norm of the vector $w^{(l)}$, $\forall l$. The regularization constants $\gamma_l \in \mathbb{R}^+$ trade-off the model complexity expressed by $w^{(l)}$ with the correct representation of the training data. $V \in \mathbb{R}^{N_{\text{Tr}} \times N_{\text{Tr}}}$ is the weighting matrix and Φ is the $N_{\text{Tr}} \times d_h$ feature matrix

$\Phi = [\varphi(x_1)^T; \dots; \varphi(x_{N_{\text{Tr}}})^T]$, where $\varphi : \mathbb{R}^d \rightarrow \mathbb{R}^{d_h}$ indicates the mapping to a high-dimensional feature space, b_l are bias terms.

After constructing the Lagrangian and solving the KKT conditions for optimality, by setting¹ $V = D^{-1}$, the following dual problem can be derived:

$$D^{-1} M_D \Omega \alpha^{(l)} = \lambda_l \alpha^{(l)} \quad (1.3)$$

where Ω is the kernel matrix with ij th entry $\Omega_{ij} = K(x_i, x_j) = \varphi(x_i)^T \varphi(x_j)$. $K : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$ indicates the kernel function. The type of kernel function to utilize depends on the specific application at hand. For instance, in the simulation results described later, three different kinds of kernels are employed, as shown in Table 1.1. The matrix D is the graph degree matrix which is diagonal with positive elements $D_{ii} = \sum_j \Omega_{ij}$, M_D is a centering matrix defined as $M_D = I_{N_{\text{Tr}}} - \frac{1}{1_{N_{\text{Tr}}}^T D^{-1} 1_{N_{\text{Tr}}}} 1_{N_{\text{Tr}}} 1_{N_{\text{Tr}}}^T D^{-1}$, the $\alpha^{(l)}$ are vectors of dual variables, $\lambda_l = \frac{N_{\text{Tr}}}{\gamma_l}$, $K : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$ is the kernel function. The projections can now be expressed as follows:

$$e_i^{(l)} = \sum_{j=1}^{N_{\text{Tr}}} \alpha_j^{(l)} K(x_j, x_i) + b_l, \quad j = 1, \dots, N_{\text{Tr}}, l = 1, \dots, k-1. \quad (1.4)$$

Problem (1.3) is related to SC with random walk Laplacian, where the kernel matrix plays the role of the similarity matrix associated to the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $v_i \in \mathcal{V}$ equals to x_i . Basically, this graph has a corresponding random walk in which the probability of leaving a vertex is distributed among the outgoing edges according to their weight: $p_{t+1} = P p_t$, where $P = D^{-1} \Omega$ indicates the transition matrix with the ij -th entry representing the probability of moving from node i to node j in one step. Under these assumptions we have an ergodic and reversible Markov chain. Furthermore, it can be shown that the stationary distribution describes the situation in which the random walker remains most of the time in the same cluster with rare jumps to the other clusters [27–29].

The cluster prototypes can be expressed in two ways:

- the projections $e_i^{(l)}$ can be binarized as $\text{sign}(e_i^{(l)})$. In fact, thanks to presence of the bias term b_l , both the $e^{(l)}$ and the $\alpha^{(l)}$ variables get automatically centred around zero. The set of the most frequent binary indicators form a code-book $\mathcal{CB} = \{c_p\}_{p=1}^k$, where each code-word is a binary word of length $k-1$ representing a cluster.
- by means of the average value of the $e_i^{(l)}$ in each cluster, as discussed in [30] where the soft KSC algorithm has been introduced.²

¹If $V = I$, problem (1.3) is equivalent to a kernel PCA formulation [22–24].

²The related Matlab code is available at: <http://www.esat.kuleuven.be/stadius/ADB/langone/softwareSKSClass.php>.

Table 1.1 Choice of the kernel function

Application	Kernel name	Mathematical expression
Vector data	RBF	$K(x_i, x_j) = \exp(-\ x_i - x_j\ _2^2/\sigma^2)$
Images	RBF_{χ^2}	$K(h^{(i)}, h^{(j)}) = \exp(-\frac{\chi_{ij}^2}{\sigma_\chi^2})$
Network data	Normalized linear	$K(x_i, x_j) = \frac{x_i^T x_j}{\ x_i\ \ x_j\ }$
Text	Normalized linear	$K(x_i, x_j) = \frac{x_i^T x_j}{\ x_i\ \ x_j\ }$
Time-series	RBF_{cd}	$K(x_i, x_j) = \exp(-\ x_i - x_j\ _{cd}^2/\sigma_{cd}^2)$

In this table RBF stands for Radial Basis Function, σ denotes the bandwidth of the kernel. The symbol $h^{(i)}$ denotes a color histogram representing the i th pixel of an image, and to compare the similarity between two histograms $h^{(i)}$ and $h^{(j)}$ the χ^2 statistical test is used [25]. Regarding time-series data, the symbol cd means correlation distance [26], and $\|x_i - x_j\|_{cd} = \sqrt{\frac{1}{2}(1 - R_{ij})}$, with R_{ij} indicating the Pearson correlation coefficient between time-series x_i and x_j .

The KSC method³ 1 is summarized in algorithm.

1.3.1.1 Out-of-Sample Extension

Given the model in its dual representation $\{\alpha^{(l)}, b_l\}$, it is possible to predict the membership of new points by computing their projections onto the eigenvectors found in the training stage:

$$e_{\text{test}}^{(l)} = \Omega_{\text{test}} \alpha^{(l)} + b_l \mathbf{1}_{N_{\text{test}}} \quad (1.5)$$

where Ω_{test} is the $N_{\text{test}} \times N$ kernel matrix evaluated using the test points with entries $\Omega_{\text{test},ri} = K(x_r^{\text{test}}, x_i)$, $r = 1, \dots, N_{\text{test}}$, $i = 1, \dots, N_{\text{Tr}}$. As for training points, the cluster indicators can be obtained in two ways:

- ECOC (Error Correcting Output Codes) decoding procedure. The score variables for test data are binarized and the memberships are assigned by comparing these indicators with the training code-book and selecting the nearest prototype based on Hamming distance.
- the test projections are assigned to the closest centroid.

³A matlab implementation of the KSC algorithm is available at: <http://www.esat.kuleuven.be/stadius/ADB/alzate/softwareKSClab.php>.

Algorithm 1: KSC algorithm [11]

Data: Training set $\mathcal{D}_{\text{Tr}} = \{x_i\}_{i=1}^{N_{\text{Tr}}}$, test set $\mathcal{D}_{\text{test}} = \{x_m^{\text{test}}\}_{m=1}^{N_{\text{test}}}$ kernel function

$K : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$ positive definite and localized ($K(x_i, x_j) \rightarrow 0$ if x_i and x_j belong to different clusters), kernel parameters (if any), number of clusters k .

Result: Clusters $\{\mathcal{A}_1, \dots, \mathcal{A}_k\}$, codebook $\mathcal{CB} = \{c_p\}_{p=1}^k$ with $\{c_p\} \in \{-1, 1\}^{k-1}$.

- 1 compute the training eigenvectors $\alpha^{(l)}$, $l = 1, \dots, k-1$, corresponding to the $k-1$ largest eigenvalues of problem (1.3)
 - 2 let $A \in \mathbb{R}^{N_{\text{Tr}} \times (k-1)}$ be the matrix containing the vectors $\alpha^{(1)}, \dots, \alpha^{(k-1)}$ as columns
 - 3 binarize A and let the code-book $\mathcal{CB} = \{c_p\}_{p=1}^k$ be composed by the k encodings of $Q = \text{sign}(A)$ with the most occurrences
 - 4 $\forall i, i = 1, \dots, N_{\text{Tr}}$, assign x_i to A_{p^*} where $p^* = \text{argmin}_p d_H(\text{sign}(\alpha_i), c_p)$ and $d_H(\cdot, \cdot)$ is the Hamming distance
 - 5 binarize the test data projections $\text{sign}(e_m^{(l)})$, $m = 1, \dots, N_{\text{test}}$, and let $\text{sign}(e_m) \in \{-1, 1\}^{k-1}$ be the encoding vector of x_m^{test}
 - 6 $\forall m$, assign x_m^{test} to A_{p^*} , where $p^* = \text{argmin}_p d_H(\text{sign}(e_m), c_p)$.
-

1.3.1.2 Model Selection

The performance of the KSC method is highly dependent on a good choice of the so called tuning parameters, like the number of clusters k and the kernel parameters (if any). For this reason, different model selection criteria have been proposed:

- *Balanced Line Fit (BLF).* It expresses how validation points belonging to the same cluster are collinear in the space of the projections. It reaches its maximum value 1 in case of well separated clusters, represented as lines in the space of the $e^{(l)}$ (see third row of Figs. 1.1 and 1.2)
- *Balanced Angular Fit or BAF* [31]. For each cluster, the sum of the cosine similarity between the validation points and the cluster prototype, divided by the cardinality of that cluster, is calculated. These similarity values are then summed up and divided by the total number of clusters.
- *Average Membership Strength abbr. AMS* [30]. The mean membership per cluster indicating the average degree of belonging of the validation points to that cluster is computed. These mean cluster memberships are then averaged over the number of clusters.
- *Modularity*, as proposed in [32, 33]. When dealing with network data, the Modularity of the validation sub-graph corresponding to a given partitioning is computed. The higher the Modularity, the strongest the community structure [34].

In Figs. 1.1 and 1.2 an example of some of these model selection criteria on a vector and a network dataset is given.

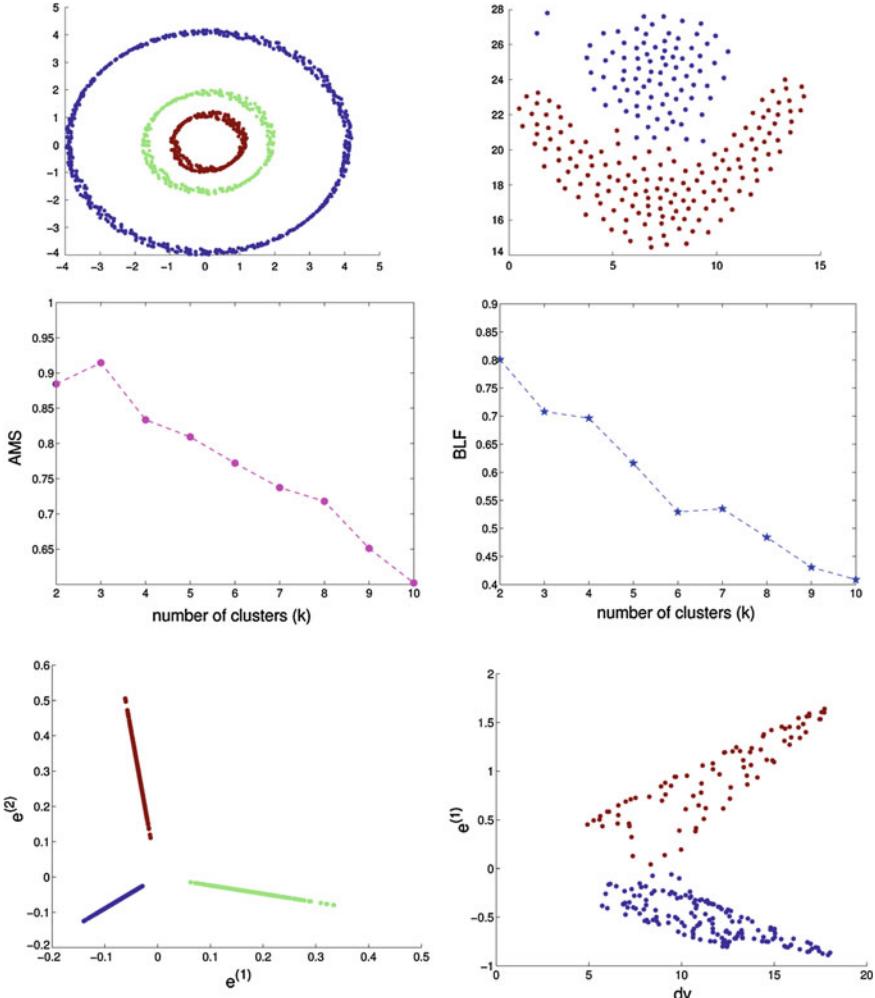
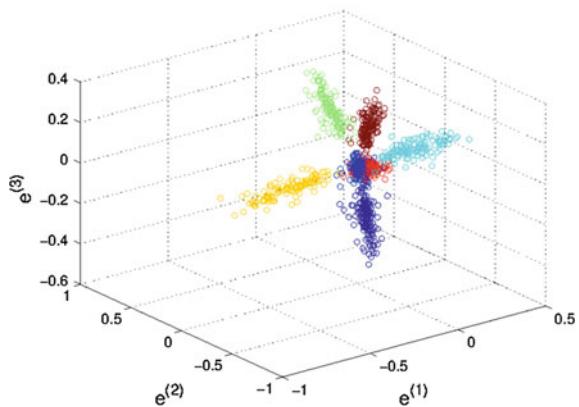
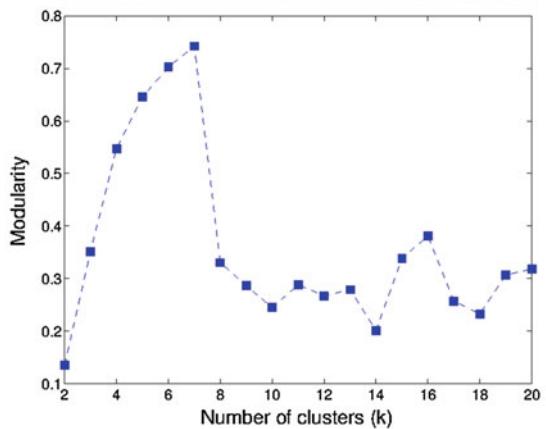
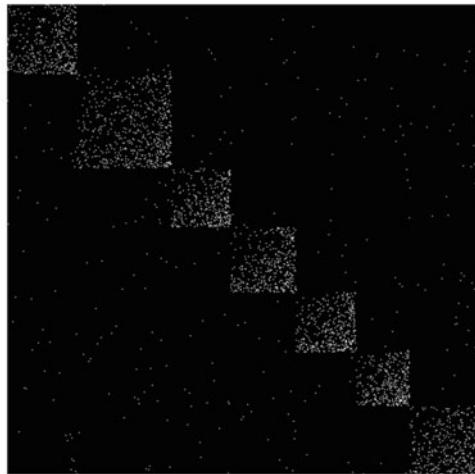


Fig. 1.1 Model selection examples. (Top) Datasets consisting of 3 clusters (left) and 2 clusters (right), in 2D. (Center) Model selection results using AMS (left) and BLF (right): the maximum is reached at $k = 3$ and $k = 2$ respectively. (Bottom) Points represented in the $[e^{(1)}, e^{(2)}]$ space (left), and the space of the first projection $e^{(1)}$ and a dummy variable $dv = \sum_{j=1}^{N_{\text{Tr}}} \Omega_{ij}^{\text{test}}$ (right). In the ideal case of well separated clusters (left) we can notice how the points belonging to one cluster lie on the same line. This line structure is less evident when a certain amount of overlap between the clusters is present, as in case of the second dataset (right)

Fig. 1.2 Modularity-based model selection.*(Top)*

Adjacency matrix of a computer-generated network consisting of 7 communities.

(Center) Model selection results using Modularity: 7 clusters are detected, corresponding to the Modularity maximum*(Bottom)* Nodes represented in the space of theprojections $[e^{(1)}, e^{(2)}, e^{(3)}]$: every cluster form a different line, which is not perfect due to some overlap between the communities of the network

1.3.2 Applications

KSC has been successfully used in a wide range of real-life applications. In [11] the algorithm is employed to perform image segmentation on pictures from the Berkeley image database [35]. The image segmentation task relates to the process of partitioning a digital image into multiple sets of pixels, such that pixels in the same group share certain visual characteristics. In the cited work only the color information is considered in order to cluster the pixels, as shown in Fig. 1.3.

The work [36] introduces a hierarchical version of KSC, which is then used for text clustering and micro-array data analysis. In [37] KSC is employed as a pre-processing step to enhance the performance of an aggregate autoregressive model for electricity power load forecasting.

The articles [38, 39] present an application of KSC to predictive maintenance. In industry, the machine status can be monitored by means of different kinds of sensors like thermometers, accelerometers and so on. Maintenance operations can then be planned in a cost efficient way if models based on sensor data are able to catch machine degradation. In the aforementioned papers, KSC is able to detect two regimes in the vibration signals collected from a packing machine. In particular, one cluster is associated to good working conditions and the other one indicates a faulty

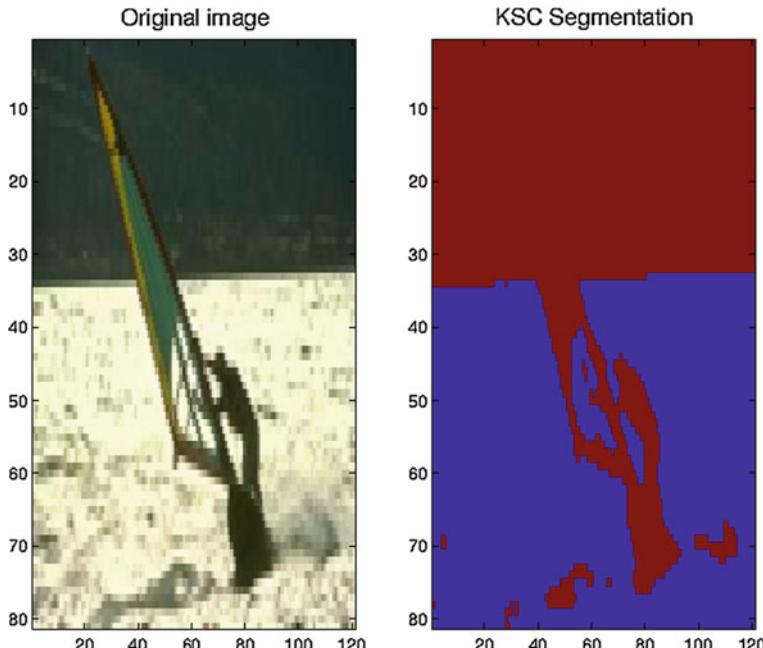


Fig. 1.3 Image segmentation. (Left) Original image (Right) Segmentation performed by KSC using only color information

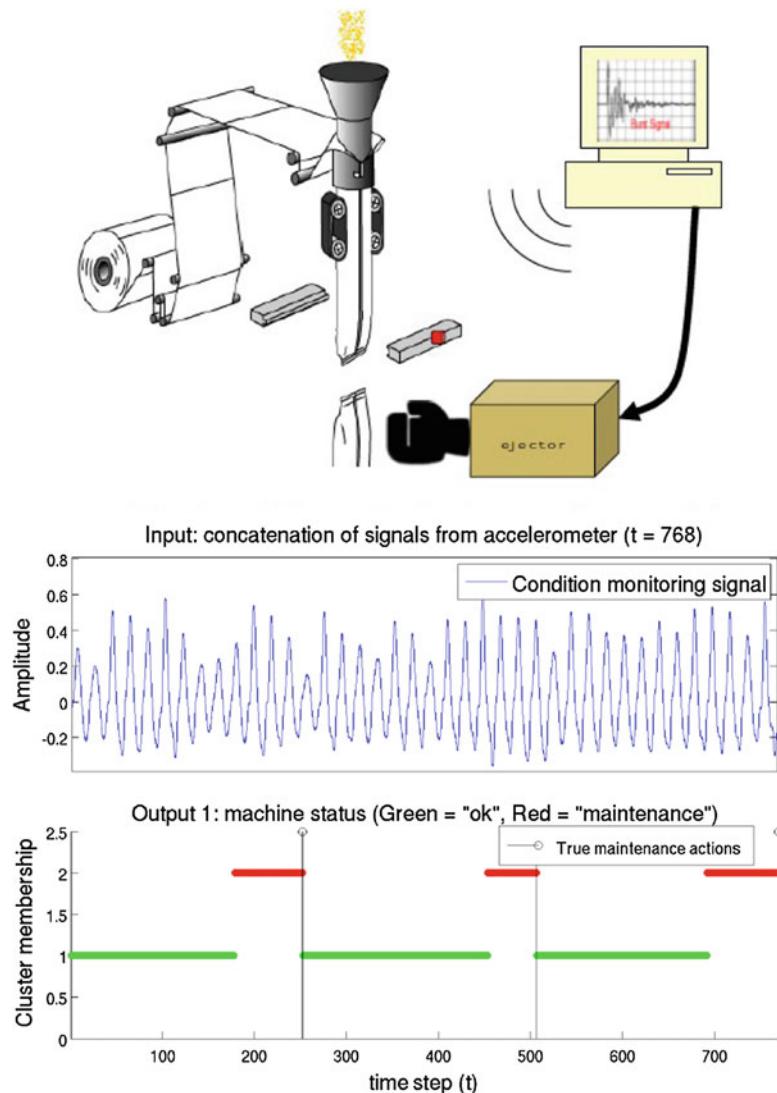


Fig. 1.4 Fault detection. (Top) Illustration of a packing machine equipped with accelerometers to measure the vibrations in the sealing jaws. (Bottom) KSC manages to infer machine's degradation based on the accelerometer signals and predicts in advance the need of maintenance (red cluster)

regime leading to maintenance (see Fig. 1.4). In this case a KSC model is trained offline, and it is successively employed online in a dynamic setting (at the run-time of the machine). This is done by means of Eq. (1.5), which allows to predict at each time the working regime of the machine.

KSC can also be considered among the state-of-the-art algorithms for community detection. Community detection refers to the problem of partitioning a complex network into clusters of nodes with high density of edges, in order to understand its structure and function. Although a profusion of algorithms are present in the literature, they are rather specific, in the sense that are based on a particular intuition. On the other hand, KSC is more flexible because in the model selection phase the user

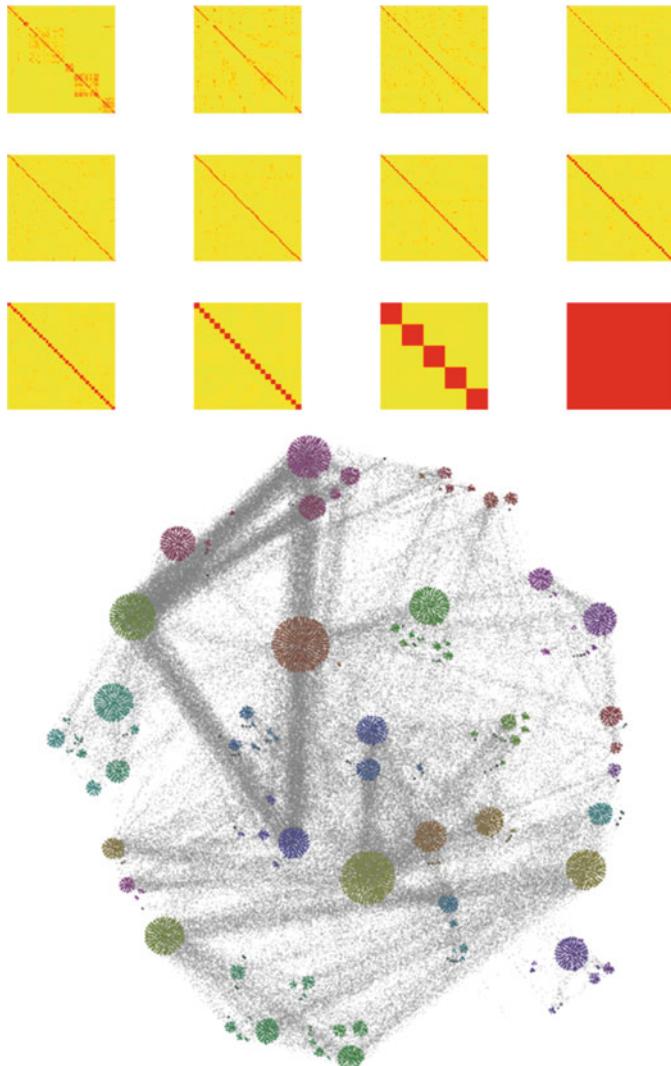


Fig. 1.5 Community detection. (Top) Hierarchical structure detected by KSC related to a Facebook network. (Bottom) Illustration of two hierarchical levels, by using the network visualization tool *Gephi* available at <http://gephi.github.io/>

can provide the desired criterion, in order to obtain a final partitioning with certain characteristics. Moreover, the out-of-sample extension allows to readily assign the membership to new nodes joining the network without using heuristics. This feature, added to the high sparsity of the majority of the real graphs, allows the method to scale to large network data even on a desktop computer [31, 40]. In Fig. 1.5 a hierarchical partitioning of a real-world network performed by KSC is depicted. This network consists of friends lists collected from survey participants using Facebook [41].

1.4 Dynamic Clustering

In many real-life applications, like text mining, genomic analysis, weather predictions etc., data are usually collected during a certain time-span. In this framework, in order to gain better insights in the phenomena of interest, dynamic clustering plays a key role. By detecting significant patterns and following their evolution, a better understanding of the system under study, in terms of the regimes it undergoes over time, can be achieved. In the next two sections we describe two kernel-based methods for dynamic clustering, namely kernel spectral clustering with memory abbr. MKSC and the incremental kernel spectral clustering (IKSC) algorithm.

1.4.1 MKSC

The MKSC model assumes that the data are given as a sequence of graphs $\mathcal{S} = \{\mathcal{G}_t = (\mathcal{V}_t, \mathcal{E}_t)\}_{t=1}^T$ over time horizon T , where t indicates the time index. The symbol \mathcal{V}_t denotes the set of nodes in the graph \mathcal{G}_t and \mathcal{E}_t the related set of edges. The graphs can represent networks or data matrices. In this last case every data point acts as a node of the graph.

MKSC is based on a constrained optimization problem where the objective function is designed to incorporate temporal smoothness, in order to cluster the current data well and to be consistent with the recent past. For each data snapshot the primal problem of the MKSC model, where N_{Tr} points are used for training, can be stated as follows [18–20]:

$$\begin{aligned} \min_{w_t^{(l)}, e_t^{(l)}, b_t^l} \quad & \frac{1}{2} \sum_{l=1}^{k-1} w_t^{(l)T} w_t^{(l)} - \frac{\gamma_t}{2N_{\text{Tr}}} \sum_{l=1}^{k-1} e_t^{(l)T} D_{\text{Mem}}^{-1} e_t^{(l)} - \nu_t \sum_{l=1}^{k-1} w_t^{(l)T} \sum_{i=1}^M w_{t-i}^{(l)} \\ \text{subject to} \quad & e_t^{(l)} = \Phi_t w_t^{(l)} + b_t^l 1_{N_{\text{Tr}}} l = 1, \dots, k-1. \end{aligned} \quad (1.6)$$

The first two terms in the objective are the same as in Eq.(1.1), i.e. they cast the clustering problem in a regularized kernel PCA formulation. The third term enforces the maximization of the correlation between the actual and the previous models, in

order to smoothen the clustering results. The subscript Mem refer to time steps $t - 1, \dots, t - M$, with M referring to the memory, that is the amount of past information. The meaning of the symbols is as follows:

- as for KSC, $e_t^{(l)}$ represent the l th binary clustering model for the N points and are referred interchangeably as projections, latent variables or score variables.
- $w_t^{(l)} \in \mathbb{R}^{d_h}$ and $b_t^{(l)}$ are the parameters of the model at time t
- $D_{\text{Mem}}^{-1} \in \mathbb{R}^{N_{\text{Tr}} \times N_{\text{Tr}}}$ is the inverse of the degree matrix $D_{\text{Mem}} = D + \sum_{i=1}^M D_{t-i}$, which is the sum of the actual degree matrix D and the M previous degree matrices
- as before Φ indicates the $N_{\text{Tr}} \times d_h$ feature matrix $\Phi = [\varphi(x_1)^T; \dots; \varphi(x_{N_{\text{Tr}}})^T]$ which expresses the relationship between each pair of points in the feature space $\varphi : \mathbb{R}^d \rightarrow \mathbb{R}^{d_h}$.
- $\gamma \in \mathbb{R}^+$ and $\nu \in \mathbb{R}^+$ are regularization constants. In particular, ν is referred as the smoothness parameter, since it constrains the actual model to resemble the old models.

The dual problem related to Eq. (1.6) becomes the following linear system:

$$(D_{\text{Mem}}^{-1} M_{D_{\text{Mem}}} \Omega_t - \frac{I}{\gamma_t}) \alpha_t^{(l)} = -\nu_t D_{\text{Mem}}^{-1} M_{D_{\text{Mem}}} \sum_{i=1}^M \Omega_{t-i} \alpha_{t-i}^{(l)} \quad (1.7)$$

where:

- Ω_t indicates the current kernel matrix with entries $\Omega_{ij} = K(x_i, x_j) = \varphi(x_i)^T \varphi(x_j)$. Ω_{t-i} captures the similarity between the objects of the current snapshot and the ones of the previous M snapshots
- $M_{D_{\text{Mem}}}$ is the centering matrix equal to $M_{D_{\text{Mem}}} = I_{N_{\text{Tr}}} - \frac{1}{1_{N_{\text{Tr}}}^T D_{\text{Mem}}^{-1} 1_{N_{\text{Tr}}}} 1_{N_{\text{Tr}}} 1_{N_{\text{Tr}}}^T D_{\text{Mem}}^{-1}$.

As in the KSC case, the MKSC algorithm allows to generate the cluster memberships for test points by projecting them into the embedding given by the dual solution vectors $\alpha_t^{(l)}$. This out-of-sample extension is described by the following formula:

$$e_t^{(l), \text{test}} = \Omega_t^{\text{test}} \alpha_t^{(l)} + \nu_t \sum_{i=1}^M \Omega_{t-i}^{\text{test}} \alpha_{t-i}^{(l)} + b_t^{(l)} 1_{N_{\text{test}}}. \quad (1.8)$$

Finally, the number of clusters, the kernel tuning parameters and the regularization constants γ_t and ν_t can be tuned by means of the smoothed counterparts of the model selection criteria introduced for KSC in Sect. 1.3.1.2, that is BLF_{Mem} , Mod_{Mem} and AMS_{Mem} . These smoothed cluster quality measures are the sum of the snapshot quality and the temporal quality. The first only measures the quality of the current clustering with respect to the current data, while the second measures the ability of the actual model to cluster the historic data. For a given cluster quality criterion CQ , its smoothed version can be defined as follows:

$$\text{CQ}_{\text{Mem}}(X_{\alpha_t}, \mathcal{G}_t) = \eta \text{CQ}(X_{\alpha_t}, \mathcal{G}_t) + (1 - \eta) \text{CQ}(X_{\alpha_t}, \mathcal{G}_{t-1}), \quad (1.9)$$

Algorithm 2: MKSC algorithm [18]

Data: Training sets $\mathcal{D} = \{x_i\}_{i=1}^{N_{\text{Tr}}}$ and $\mathcal{D}_{\text{old}} = \{x_i^{\text{old}}\}_{i=1}^{N_{\text{Tr}}}$, test sets $\mathcal{D}^{\text{test}} = \{x_m^{\text{test}}\}_{m=1}^{N_{\text{test}}}$ and $\mathcal{D}_{\text{old}}^{\text{test}} = \{x_m^{\text{test,old}}\}_{m=1}^{N_{\text{test}}}$, $\alpha_{\text{old}}^{(l)}$, where the term *old* refers to time-steps $i - 1, \dots, i - M$, positive definite kernel function $K : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$ such that $K(x_i, x_j) \rightarrow 0$ if x_i and x_j belong to different clusters, kernel parameters (if any), number of clusters k , regularization constants γ and v .

Result: Clusters $\{\mathcal{C}_1^t, \dots, \mathcal{C}_k^t\}$, cluster codeset $\mathcal{CB} = \{c_p\}_{p=1}^k$, $c_p \in \{-1, 1\}^{k-1}$.

```

1 if  $t == 1$  then
2   | Initialization by using KSC.
3 else
4   | Compute the solution vectors  $\alpha^{(l)}$ ,  $l = 1, \dots, k - 1$ , related to the linear system
      described by Eq. (1.7):  $(D_{\text{Mem}}^{-1} M D_{\text{Mem}} \Omega_t - \frac{I}{\gamma}) \alpha_t^{(l)} = -v_t D_{\text{Mem}}^{-1} M D_{\text{Mem}} \sum_{i=1}^M \Omega_{t-i} \alpha_{t-i}^{(l)}$ 
5   | Binarize the solution vectors:  $\text{sign}(\alpha_i^{(l)})$ ,  $i = 1, \dots, N_{\text{Tr}}$ ,  $l = 1, \dots, k - 1$ , and let
       $\text{sign}(\alpha_i) \in \{-1, 1\}^{k-1}$  be the encoding vector for the training data point  $x_i$ .
6   | Count the occurrences of the different encodings and find the  $k$  encodings with most
      occurrences. Let the codeset be formed by these  $k$  encodings:  $\mathcal{CB} = \{c_p\}_{p=1}^k$ , with
       $c_p \in \{-1, 1\}^{k-1}$ .
7   |  $\forall i$ , assign  $x_i$  to  $C_{p^*}$  where  $p^* = \operatorname{argmin}_p d_H(\text{sign}(\alpha_i), c_p)$  and  $d_H(\cdot, \cdot)$  is the Hamming
      distance.
8   | Binarize the test data projections  $\text{sign}(e_m^{(l)})$ ,  $m = 1, \dots, N_{\text{test}}$ ,  $l = 1, \dots, k - 1$  and let
       $\text{sign}(e_m) \in \{-1, 1\}^{k-1}$  be the encoding vector of  $x_m^{\text{test}}$ ,  $m = 1, \dots, N_{\text{test}}$ .
9   |  $\forall m$ , assign  $x_m^{\text{test}}$  to  $C_{p^*}^t$  using an ECOC decoding scheme, i.e.
       $p^* = \operatorname{argmin}_p d_H(\text{sign}(e_m), c_p)$ .
10 end
```

where X_{α_t} means the cluster indicator matrix calculated by using the current solution vectors $\alpha_t^{(l)}$. The parameter η reflects the emphasis given to the snapshot quality and the temporal smoothness, respectively. Finally, a summary of the MKSC technique⁴ is provided in algorithm 2.

1.4.1.1 Applications

In this section two applications of the MKSC method are described.

The first application concerns community detection of an evolving network named *RealityNet*. This dataset records the proximity of some students and staff members from two different departments in MIT [42]. It is constructed on users whose cell-phones periodically scan for nearby phones via Blue-tooth at five minutes intervals. The similarity between two users is related to the number of intervals where they were in physical proximity. Each graph snapshot is a weighted network corresponding to 1 week activity, and a total of 46 snapshots covering the entire 2005 academic year are present. The people part of this experiment are in total 94, but not all of

⁴A software package implemented in Matlab is available at: <http://www.esat.kuleuven.be/stadius/ADB/langone/softwareMKSClab.php>.

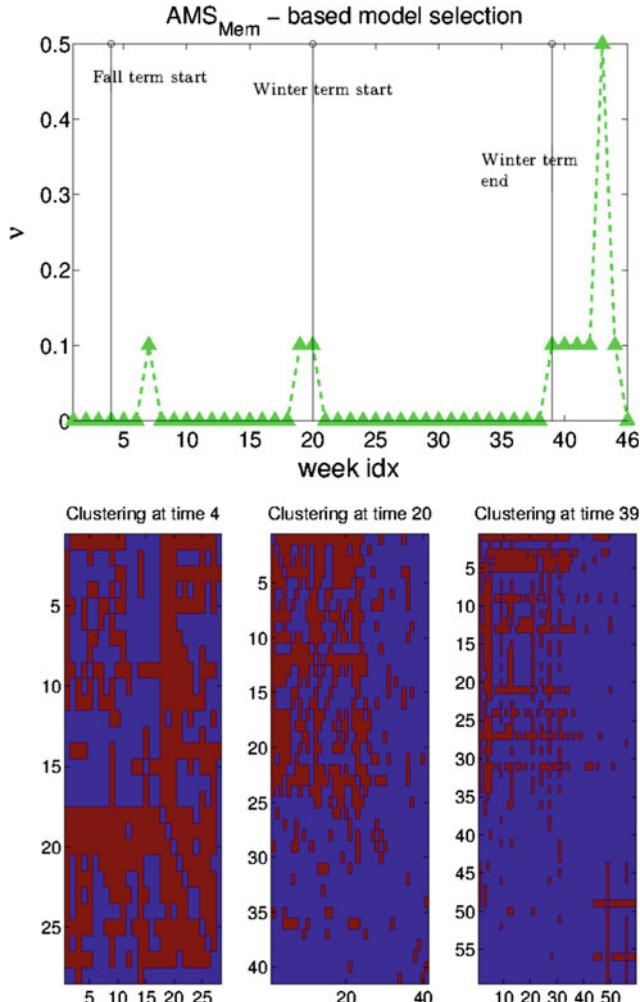


Fig. 1.6 The *RealityNet* experiment. (Top) Tuning of the regularization constant ν by means of the AMS_{Mem} criterion. Some peaks are present around important dates which are labelled in the plot, where presumably a big rearrangement of the network occurs. (Bottom) Community structure in the *RealityNet* network detected by MKSC at time steps 4, 20 and 39

them are present in every week. The smallest network comprises 21 people and the largest has 88 nodes. In Fig. 1.6 the results obtained by MKSC are visualized. At the bottom side some adjacency matrices representing the detected structure in a number of time-steps are depicted. The results show two clusters shrinking and expanding over time. These findings are in agreement with the ground-truth, namely the affiliations of each participant as students at the Sloan business school or co-workers who work in the same building. At the top side of Fig. 1.6 the tuning of the smoothness

parameter ν_t is depicted. The regularization constant has some small peaks around important dates like beginning of fall and winter term and end of winter term. This outcome can be explained by considering that, when there is a significant change in the data, the memory effect must activate to smoothen the clustering results.

The second application is related to a text mining problem. We analyse the *RCV1-5topic* dataset, which is constructed from a subset of the *Reuters RCV1* corpus [43]. There are in total 10 116 news articles covering a period of 7 months from September 1996 to March 1997. The data are divided into 28 snapshots, and each of them contains news articles related to a one week period. In Figs. 1.7, 1.8 and 1.9 we show the clustering results for three particular weeks by means of word clouds.⁵ Although there is a large amount of overlap between the clusters, we can notice how in the first week of September 1996 (Fig. 1.7) the cluster number 1 (top) comprises mainly words related to weather, the second cluster (center) is more related to medicine and economics and the third one concerns mostly sport articles. In the second week (end of November 1996) only two cluster were detected: the first one (top center of Fig. 1.8) contains words regarding weather, medicine and sport, while the second one concerns mainly scientific news (physics, energy etc.). Finally, in the third week (end of March 1997) we can see how the first cluster shown on the top of Fig. 1.9 is related mainly to medicine, the second one (center) comprises words concerning weather and politics, the third one concerns mostly economy and epidemiology.

1.4.2 IKSC

The IKSC method is intended to cluster data streams. A data stream is an ordered sequence of instances which changes continuously and rapidly. Examples of data streams include computer network traffic, phone conversations, ATM transactions, web searches, sensor data. In many data stream mining applications, the goal is to predict the class/cluster of new instances in the data stream given some knowledge about the class/cluster membership of the previous instances. In this context, incremental learning techniques are often applied to cope with structural changes and non-stationarities. In incremental k-means [17] the algorithm at time t is initialized with the centroids found at time $t - 1$, the algorithms described in [44, 45] aim at analysing massive datasets by using limited memory and a single scanning of the data, the techniques introduced in [46–48] have the objective to apply dynamic updates to the cluster prototypes when new data points arrive. In [8, 9], the authors propose some incremental eigenvalue solutions to continuously update the initial eigenvectors found by spectral clustering.

The IKSC algorithm works in a similar fashion. In the initialization phase an initial KSC model is constructed. Then, in the online stage the model is expressed only in terms of the centroids in the eigenspace, and the training set is formed by the centroids in the original input space. The cluster memberships for new points belonging to the

⁵The software to generate the word clouds visualization is available at: <http://www.wordle.net/>.



Fig. 1.7 Text mining results, week 1 Results of the MKSC algorithm applied to *Reuters RCV1* corpus related to week number 1, that is the first week of September 1996. Although a large amount of overlap between the three detected clusters is present, some clusters concern mainly a certain kind of topics compared to the others. In particular, cluster number 1 (*top*) comprises many words related to weather like hurricane, thunderstorms etc., the second cluster (*center*) is more related to health and economics, and the third one (*bottom*) concerns mostly sports articles

data stream are computed by means of the euclidean distance between their projection in the eigenspace and the centers. In this way it is possible to continuously update the model in response to the new data. In order to calculate the projection in the eigenspace for every new point, the second KKT condition for optimality related to the KSC optimization problem (1.1), which links the eigenvectors and the score variables for training data, can be exploited:



Fig. 1.8 Text mining results, week 12 Results of the MKSC algorithm applied to *Reuters RCV1* corpus related to week number 12, that is the last week of November 1996. Two clusters were detected concerning weather, medicine and sport (*top*), and scientific news (*bottom*)

$$\alpha_{\text{test}}^{(l)} = \frac{1}{\lambda_l} D_{\text{test}}^{-1} e_{\text{test}}^{(l)} \quad (1.10)$$

with $D_{\text{test}}^{-1} = \text{diag}(1/\deg(x_1^{\text{test}}), \dots, 1/\deg(x_{N_{\text{test}}}^{\text{test}})) \in \mathbb{R}^{N_{\text{test}}} \times \mathbb{R}^{N_{\text{test}}}$ indicating the inverse degree matrix for test data. The out-of-sample eigenvectors $\alpha_{\text{test}}^{(l)}$ represent the model-based eigen-approximation with the same properties as the original eigenvectors $\alpha^{(l)}$ for training data. With the term eigen-approximation we mean that these eigenvectors are not the solution of an eigenvalue problem, but they are estimated by means of a model built during the training phase of KSC [49]. To summarize, once one or several new points belonging to a data-stream are collected, we update the IKSC model as follows:

- calculate the out-of-sample extension using Eq. (1.5), where the training points x_i are the centroids in the input space C_1, \dots, C_k , and the $\alpha^{(l)}$ are the centroids in the eigenspace $C_1^\alpha, \dots, C_k^\alpha$
 - calculate the out-of-sample eigenvectors by means of Eq. (1.10)
 - assign the new points to the closest centroids in the eigenspace
 - update the centroids in the eigenspace
 - update the centroids in the input space

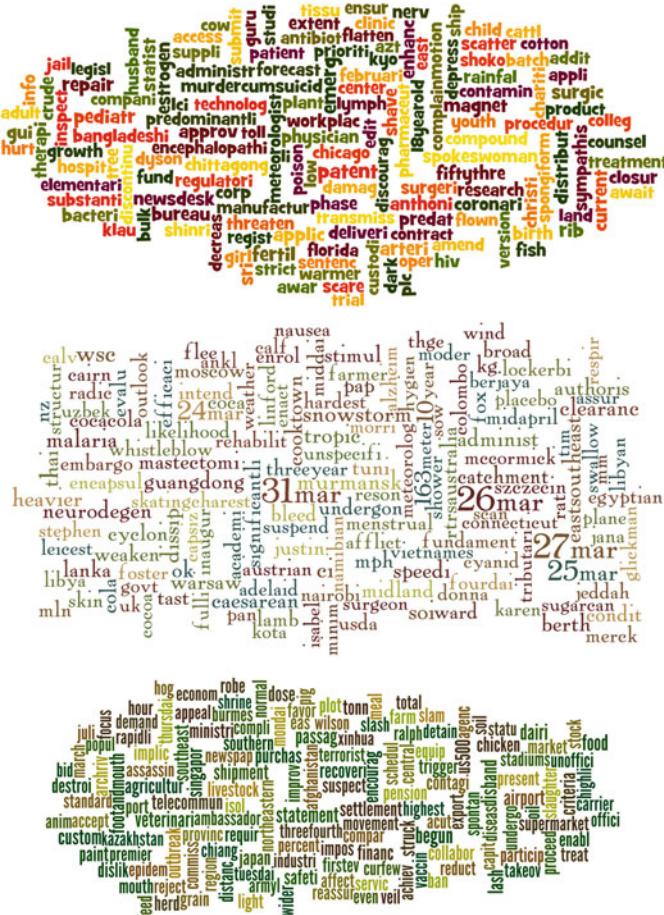


Fig. 1.9 Text mining results, week 28 Results of the MKSC algorithm applied to *Reuters RCV1* corpus related to week number 28, that is the last week of March 1997. The three clusters comprise articles related to medicine (*top*), weather and politics (*center*), economy and epidemiology (*bottom*)

In this way the initial $\alpha^{(l)}$ provided by KSC are changed over time to model the non-stationary behaviour of the system. The adaptation to non-stationarities relates to identifying changes in the number of clusters occurring over time by:

- dynamically creating a new cluster if necessary. For every new point the related degree d_i^{test} is calculated. If $d_i^{\text{test}} < \varepsilon$ where ε is a user-defined threshold, it means that the point is dissimilar to the actual centroids. Therefore it becomes the centroid of a new cluster and it is added to the model. The threshold ε is data-dependent, and can be chosen before processing the data stream based on the degree distribution of the test kernel matrix, when considering as training set the cluster prototypes in the input space.

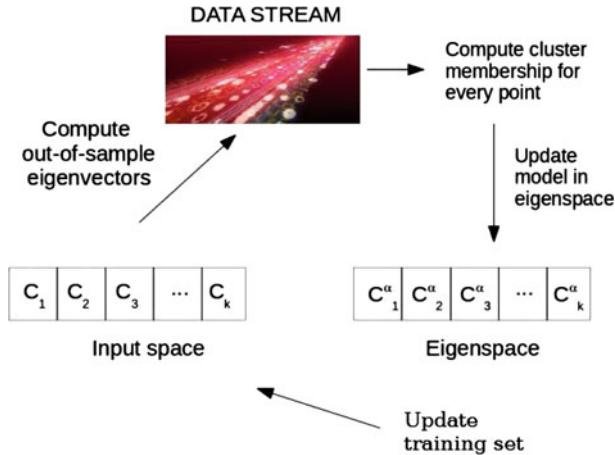


Fig. 1.10 IKSC update scheme After the initialization phase, whenever a new instance of a data stream is processed, both the training set and the model (i.e. the cluster centers in the eigenspace) are updated

- merging two centroids into one center if they become too similar. In particular, the similarity between two centroids is computed as the cosine similarity in the eigenspace, and two centroids are merged if this similarity is greater than 0.5.

A schematic visualization of the IKSC procedure.⁶ is sketched in Fig. 1.10

1.4.2.1 Applications

Here we describe an application of the IKSC technique to time-series clustering. We analyse the PM₁₀ concentrations registered by 259 background stations (located in Belgium, The Netherlands, Germany and Luxembourg) during a heavy pollution episode occurred between January 20th, 2010 and February 1st, 2010. The experts attributed this episode to the import of particle matter originating in Eastern Europe, due to strong winds.

An initial model is constructed by considering the data related to the first 96 hours: only 2 clusters are detected. The remaining data is then processed using a moving window approach, i.e. the data-set at time t corresponds to the PM₁₀ concentrations measured from time $t - 96$ to time t . After some time the IKSC model creates a new cluster, as depicted in Fig. 1.11. Later on these three clusters evolve until a merge of two of them occurs at time step $t = 251$. The new cluster (represented in blue) comprise stations which are mainly concentrated in the Northern region of Germany. Moreover, the creation occurs at time step $t = 143$, when the window

⁶A matlab implementation of the IKSC algorithm is available at: <http://www.esat.kuleuven.be/stadius/ADB/langone/softwareIKSCLab.php>.

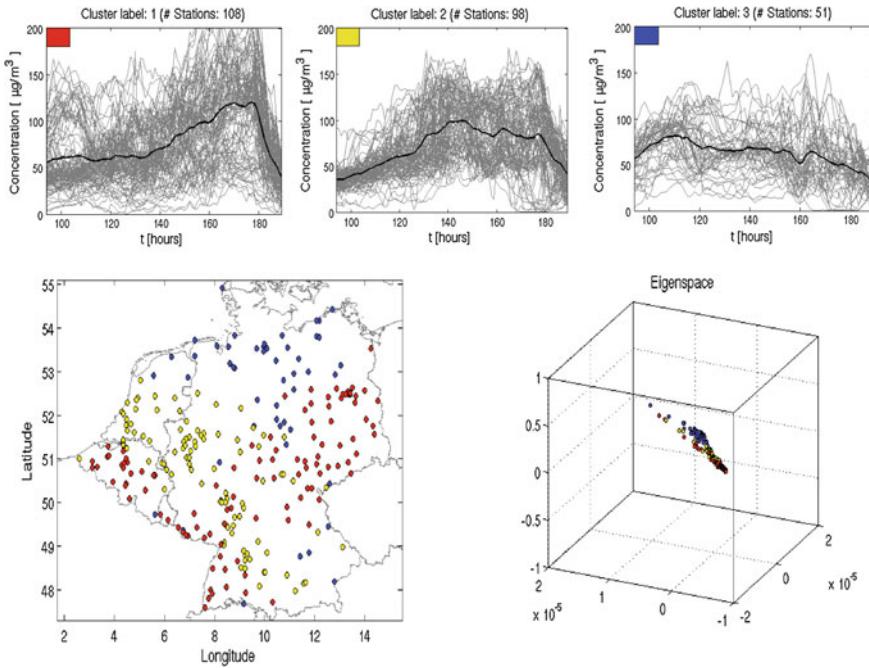


Fig. 1.11 PM₁₀ clusters after creation. *Top* clustered PM₁₀ time-series after the creation of a new cluster. *Bottom left* Spatial distribution of the clusters over Belgium, Netherlands, Luxembourg and Germany. The new cluster comprises stations located in the North-East part of Germany, which is the area where the pollutants coming from Eastern Europe started to spread during the heavy pollution episode of January 2010. *Bottom right* data in the space spanned by the eigenvectors $\alpha^{(1)}$ and $\alpha^{(2)}$

describes the start of the pollution episode in Germany. Afterwards, the new cluster starts expanding in direction South-West and then disappears. Basically, IKSC is detecting the arrival of the pollution episode originated in Eastern Europe and driven by the wind toward the West.

1.5 Concluding Remarks

In this chapter we have discussed two algorithms designed in a kernel-based framework able to cluster dynamic data, namely MKSC and IKSC, in relation to kernel spectral clustering. The former assumes that the data are provided as a sequence of matrices over time, and makes use of a temporal smoothness assumption in order to properly model the long-term trend of the cluster structure, while disregarding short-term fluctuations due to noise. On the other hand, IKSC is mainly meant to cluster data streams, where an initial model needs to be promptly updated in response to new

data in order to cope with non-stationary data distributions. Both models are based on KSC, which is also described in the beginning of the chapter. The KSC algorithm, although it is a static method, can also be used in a dynamic setting by means of the out-of-sample extension property, as explained in Sect. 1.3.2 for the fault detection application. Finally, beyond discussing previous results related to community detection, image segmentation and time-series clustering, we have also presented a new application related to dynamic text mining.

Acknowledgments EU: The research leading to these results has received funding from the European Research Council under the European Union’s Seventh Framework Programme (FP7/2007-2013) / ERC AdG A-DATADRIVE-B (290923). This chapter reflects only the authors’ views, the Union is not liable for any use that may be made of the contained information. Research Council KUL: GOA/10/09 MaNet, CoE PFV/10/002 (OPTEC), BIL12/11T; PhD/Postdoc grants. Flemish Government: FWO: projects: G.0377.12 (Structured systems), G.088114N (Tensor based data similarity); PhD/Postdoc grants. IWT: projects: SBO POM (100031); PhD/Postdoc grants. iMinds Medical Information Technologies SBO 2014. Belgian Federal Science Policy Office: IUAP P7/19 (DYSCO, Dynamical systems, control and optimization, 2012–2017.)

References

1. Chung, F.R.K.: Spectral Graph Theory. American Mathematical Society, Providence (1997)
2. Shi, J., Malik, J.: Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **22**(8), 888–905 (2000)
3. Ng, A.Y., Jordan, M.I., Weiss, Y.: On spectral clustering: analysis and an algorithm. *Advances in Neural Information Processing Systems*, vol. 14, pp. 849–856. MIT Press, Cambridge (2002)
4. von Luxburg, U.: A tutorial on spectral clustering. *Stat. Comput.* **17**(4), 395–416 (2007)
5. Fowlkes, C., Belongie, S., Chung, F., Malik, J.: Spectral grouping using the Nyström method. *IEEE Trans. Pattern Anal. Mach. Intell.* **26**(2), 214–225 (2004)
6. Williams, C.K.I., Seeger, M.: Using the Nyström method to speed up kernel machines. *Advances in Neural Information Processing Systems*, vol. 13. MIT Press, Cambridge (2001)
7. Lin, F., Cohen, William, W.: Power iteration clustering. In: ICML, pp. 655–662 (2010)
8. Ning, H., Xu, W., Chi, Y., Gong, Y., Huang, T.S.: Incremental spectral clustering by efficiently updating the eigen-system. *Pattern Recognit.* **43**(1), 113–127 (2010)
9. Charanpal, D., Romaric, G., Stephan, C.: Efficient eigen-updating for spectral graph clustering (2013). [arXiv:1301.1318](https://arxiv.org/abs/1301.1318)
10. Frederix, K., Van Marc, B.: Sparse spectral clustering method based on the incomplete Cholesky decomposition. *J. Comput. Appl. Math.* **237**(1), 145–161 (2013)
11. Alzate, C., Suykens, J.A.K.: Multiway spectral clustering with out-of-sample extensions through weighted kernel PCA. *IEEE Trans. Pattern Anal. Mach. Intell.* **32**(2), 335–347 (2010). (ESAT-SISTA, K.U. Leuven)
12. Suykens, J.A.K., Van Gestel, T., De Brabanter, J., De Moor, B., Vandewalle, J.: Least Squares Support Vector Machines. World Scientific, Singapore (2002)
13. Chakrabarti, D., Kumar, R., Tomkins, A.: Evolutionary clustering. In: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery And Data Mining, pp. 554–560. ACM, New York (2006)
14. Chi, Y., Song, X., Zhou, D., Hino, K., Tseng, B.L.: Evolutionary spectral clustering by incorporating temporal smoothness. In: KDD’07, pp. 153–162 (2007)
15. Xu, K.S., Kliger, M., Hero, III., Alfred, O.: Adaptive evolutionary clustering. *Data Min. Knowl. Discov.* 1–33 (2013)

16. Mucha, P.J., Richardson, T., Macon, K., Porter, M.A., Onnela, J.P.: Community structure in time-dependent, multiscale, and multiplex networks. *Science* **328**(5980), 876–878 (2010)
17. Chakraborty, S., Nagwani, N.K.: Analysis and study of incremental K-means clustering algorithm. *High Perform. Archit. Grid Comput.* **169**, 338–341 (2011)
18. Rocco, L., Carlos, A., Suykens, J.A.K.: Kernel spectral clustering with memory effect. *Phys. A: Stat. Mech. Appl.* **392**(10), 2588–2606 (2013)
19. Rocco, L., Suykens, J.A.K.: Community detection using kernel spectral clustering with memory. *J. Phys. Conf. Ser.* **410**(1), 012100 (2013)
20. Langone, R., Mall, R., Suykens, J.A.K.: Clustering data over time using kernel spectral clustering with memory. In: *SSCI (CIDM)* (2014)
21. Langone, R., Agudelo, O.M., De Moor, B., Suykens, J.A.K.: Incremental kernel spectral clustering for online learning of non-stationary data. *Neurocomputing* **139**, 246–260 (2014)
22. Suykens, J.A.K., Van Gestel, T., Vandewalle, J., De Moor, B.: A support vector machine formulation to PCA analysis and its kernel version. *IEEE Trans. Neural Netw.* **14**(2), 447–450 (2003)
23. Schölkopf, B., Smola, A.J., Müller, K.R.: Nonlinear component analysis as a kernel eigenvalue problem. *Neural Comput.* **10**, 1299–1319 (1998)
24. Mika, S., Schölkopf, B., Smola, A.J., Müller, K.R., Scholz, M., Rätsch, G.: Kernel PCA and de-noising in feature spaces. *Advances in Neural Information Processing Systems*, vol. 11. MIT Press, Cambridge (1999)
25. Puzicha, J., Hofmann, T., Buhmann, J.: Non-parametric similarity measures for unsupervised texture segmentation and image retrieval. *Comput. Vis. Pattern Recognit.* 267–272 (1997)
26. Warren, L.T.: Clustering of time series data—a survey. *Pattern Recognit.* **38**(11), 1857–1874 (2005)
27. Meila, M., Shi, J.: A random walks view of spectral segmentation. In: *Artificial Intelligence and Statistics AISTATS* (2001)
28. Meila, M., Shi, J.: Learning segmentation by random walks. *Advances in Neural Information Processing Systems*, vol. 13. MIT Press, Cambridge (2001)
29. Delvenne, J.C., Yaliraki, S.N., Barahona, M.: Stability of graph communities across time scales. *Proc. Natl. Acad. Sci.* **107**(29), 12755–12760 (2010)
30. Langone, R., Mall, R., Suykens, J.A.K.: Soft Kernel spectral clustering. In: *Proceedings of the International Joint Conference on Neural Networks (IJCNN 2013)*, pp. 1028–1035 (2013)
31. Mall, R., Langone, R., Suykens, J.A.K.: Kernel spectral clustering for big data networks. *Entropy (Special Issue on Big Data)* **15**(5), 1567–1586 (2013)
32. Langone, R., Alzate, C., Suykens, J.A.K.: Modularity-based model selection for kernel spectral clustering. In: *Proceedings of the International Joint Conference on Neural Networks (IJCNN 2011)*, pp. 1849–1856 (2011)
33. Langone, R., Alzate, C., Suykens, J.A.K.: Kernel spectral clustering for community detection in complex networks. In: *Proceedings of the International Joint Conference on Neural Networks (IJCNN 2012)*, pp. 2596–2603 (2012)
34. Newman, M.E.J.: Modularity and community structure in networks. *Proc. Natl. Acad. Sci. USA* **103**(23), 8577–8582 (2006)
35. Martin, D., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: *Proceedings of the 8th International Conference on Computer Vision*, vol. 2, pp. 416–423 (2001)
36. Alzate, C., Suykens, A.K.J.: Hierarchical kernel spectral clustering. *Neural Netw.* **35**, 21–30 (2012)
37. Alzate, C., Sinn, M.: Improved electricity load forecasting via kernel spectral clustering of smart meters. In: *ICDM*, pp. 943–948 (2013)
38. Langone, R., Alzate, C., De Ketelaere, B., Suykens, J.A.K.: Kernel spectral clustering for predicting maintenance of industrial machines. In: *IEEE Symposium Series on Computational Intelligence and data mining SSCI (CIDM) 2013*, pp. 39–45 (2013)

39. Langone, R., Alzate, C., De Ketelaere, B., Vlasselaer, J., Meert, W., Suykens, J.A.K.: LS-SVM based spectral clustering and regression for predicting maintenance of industrial machines. *Eng. Appl. Artif. Intell.* **37**, 268–278 (2015)
40. Mall, R., Langone, R., Suykens, J.A.K.: Multilevel hierarchical kernel spectral clustering for real-life large scale complex networks. *PLoS ONE. Public Libr. Sci.* **9**(6), e99966 (2014)
41. McAuley, J., Jure, L.: Discovering social circles in ego networks. *TKDD* **8**(1), 4 (2014)
42. Eagle, N., Pentland, A.S., Lazer, D.: Inferring social network structure using mobile phone data. *PNAS* **106**(1), 15274–15278 (2009)
43. Derek, G., Padraig C.: Spectral co-clustering for dynamic bipartite graphs. In: Proceedings of the 1st Workshop on Dynamic Networks and Knowledge Discovery, Barcelona, Spain (2010)
44. Guha, S., Meyerson, A., Mishra, N., Motwani, R., O'Callaghan, L.: Clustering data streams: theory and practice. *IEEE Trans. Knowl. Data Eng.* **15**(3), 515–528 (2003)
45. Aggarwal, C.C., Han, J., Wang, J., Yu, P.S.: A framework for clustering evolving data streams. In: Proceedings of the 29th International Conference on Very Large Data Bases, vol. 29, VLDB '03, pp. 81–92 (2003)
46. Can, F.: Incremental clustering for dynamic information processing. *ACM Trans. Inf. Syst.* **11**(2), 143–164 (1993)
47. Gupta, C., Grossman, R.L.: GenIc: a single-pass generalized incremental algorithm for clustering. In: SDM. SIAM (2004)
48. Ning, H., Xu, W., Chi, Y., Gong, Y., Huang, T.S.: Incremental spectral clustering with application to monitoring of evolving blog communities. In: SDM (2007)
49. Alzate, C., Suykens, J.A.K.: Out-of-Sample eigenvectors in kernel spectral clustering. In: Proceedings of the International Joint Conference on Neural Networks (IJCNN 2011), pp. 2349–2356 (2011)

Chapter 2

Community Detection in Bipartite Networks: Algorithms and Case studies

Taher Alzahrani and K.J. Horadam

Abstract There is increasing motivation to study bipartite complex networks as a separate category and, in particular, to investigate their community structure. We outline recent work in the area and focus on two high-performing algorithms for unipartite networks, the modularity-based *Louvain* and the flow-based *Infomap*. We survey modifications of modularity-based algorithms to adapt them to the bipartite case. As *Infomap* cannot be applied to bipartite networks for theoretical reasons, our solution is to work with the primary projected network. We apply both algorithms to four projected networks of increasing size and complexity. Our results support the conclusion that the clusters found by *Infomap* are meaningful and better represent ground truth in the bipartite network than those found by *Louvain*.

2.1 Introduction

A very large number of clustering algorithms is available for community detection in networks. These algorithms try to identify subgraphs (often called communities, clusters or modules) which are more tightly connected internally, according to a particular measurable rule, than they are connected to the rest of the network. The practical aim is to derive a coarse-grain picture of a real large-scale network which will aid understanding of its hierarchical structure. However, there may not be a strong correlation between the clusters found by an algorithm and the ground truth of hierarchical structure within the network, since real-world community formation may be a result of many interacting and potentially unmeasurable rules. In any case, the ground truth in a real network may not be directly discernable by virtue of the network's size and complexity. Thus we would like to have some confidence in the meaningfulness of the optimal partition arrived at by a clustering algorithm.

T. Alzahrani · K.J. Horadam (✉)

School of Mathematical and Geospatial Sciences, RMIT University, Melbourne,
VIC 3001, Australia
e-mail: kathy.horadam@rmit.edu.au

T. Alzahrani
e-mail: taher.alzahrani@rmit.edu.au

This issue becomes more complicated for bipartite (or, more generally, multiparite) networks. Simple, or unipartite, networks are the typical framework for complex network study. However, many complex networks can best be described as bipartite [36]. In a bipartite network, the node set consists of two disjoint sets of nodes such that links between nodes may occur only if the nodes belong to different sets. Examples of bipartite networks are citation networks between authors and published papers in academia, recommendation systems in online purchasing, protein interaction networks in biological science and movie-actor networks in social networks.

Obviously, every bipartite network can be treated as unipartite by ignoring the node partition, but in the last few years, there has been increasing motivation to analyse bipartite networks as a separate network category, and in particular to investigate their community structure.

Usually one set of nodes in a bipartite network, the *primary set* P , is of more interest for a particular purpose than the other, the *secondary set* S . In this case, P may be treated as the node set of a unipartite *projection* network, whose edges are derived from linking information in the bipartite network. A battery of unipartite clustering algorithms may then be applied directly to the projection. The rôles of the two node sets can be switched for different applications.

Many real networks arise naturally as projections of bipartite networks. It has also been argued [20] that *every* complex network is a projection of a bipartite network constructed from its node set (as P) and from a set of cliques that it contains (as S), and that this bipartite model explains many of the network's main properties.

There are different ways of defining the edges in a projection on P . Furthermore, the structure of the projection on P will depend on S in important ways: in [32] it is shown that the degree distribution of the projection on P depends very strongly on the degree distribution of S .

So there are really two approaches to identifying clusters in a bipartite network: the first, and more common, is when our real interest is in community structure within the primary node set P ; and the second is when our real interest is in bipartite communities within the whole network.

In this chapter we focus on the first approach. We outline recent work on community detection algorithms for unipartite networks and how they have been adapted, or else cannot be applied, to bipartite networks. We apply the two highest-performing algorithms to several projected networks. Our results support the conclusion that the clusters found by the flow-based algorithm *Infomap* better represent ground truth in the bipartite network than those found by the best modularity-based algorithm *Louvain*.

The chapter is organised as follows. In the next Sect. 2.2 we give an overview of clustering algorithms which are used for partitioning nodes into non-overlapping communities in a large and complex unipartite network. For unipartite networks, two approaches to community detection have been very popular, one based on modelling the clustering structure and one based on extracting it from flow calculations on the network. The best algorithms to cluster very large networks using each approach, compared using the LFR benchmark datasets [27], are now referred to as the *Louvain* algorithm [9] and the *Infomap* algorithm [47].

Section 2.3 surveys how these and other algorithms have been modified for the important class of bipartite networks. As Infomap cannot be directly applied to bipartite networks, in Sect. 2.4 we describe its application to the network found by weighted projection onto the primary node set P . In Sect. 2.5 we look further at the critical issue of whether the clusters found by Infomap in the weighted projection network make sense: that is, whether or not they represent some sort of ground truth. We present several case studies to support the proposition that they do. Finally, in Sect. 2.6 we describe our intended solution to adapt Infomap to bipartite networks and propose a list of further research questions.

2.2 Community Detection Algorithms

In complex networks a community (or cluster, or module) is a fundamental qualitative concept for which there is still no single accepted definition. It may be a node based idea, as we use here, or an edge based one.

In a node based definition, a cluster is a set of nodes which connect more to each other than to other nodes of the network, based on the idea that they share the same resources or have similar properties. This kind of definition is widely accepted and used. A well-known quality function that evaluates clusters based on this idea is modularity [35].

On the other hand, in an edge based definition, a cluster is a group of edges rather than of nodes [1, 13]. The classification of edges into groups is based on their similarity through sharing nodes of the network. This definition is useful in dealing with overlapping communities, where each node inherits membership from all its incident edges and can belong to multiple communities according to the similarity between these edges.

Such different definitions lead to rapid evolution of a vast number of cluster detection techniques [15]. From our point of view, the choice of definition depends on the context and application requirements for a particular network. For example, on the World Wide Web (WWW) a cluster can be looked at as information or as physical links and routers connecting to each other. Scientific collaborations can be classified as clusters of scientists, clusters of papers or both. Social network clusters can be defined as people relating to each other or as interests that are shared by a group of people.

In this section we will first establish some basic concepts about clusters and comparison of partitions, then describe the LFR benchmarks for testing performance of community detection algorithms. We follow with a description of modularity-based algorithms and the problem of the resolution limit and then conclude by outlining flow-based algorithms.

We will use the following notation throughout: in a network $G = (V, E)$ with node set V and edge set $E \subseteq V \times V$, we set $n = |V|$, $m = |E|$ and let $A = [A_{ij}]$ represent the network's adjacency matrix. (If the network has multiple edges then

A_{ij} is the number of edges from node i to node j .) Node i will have degree k_i , and we observe that $k_i = \sum_j A_{ij}$. The complete graph or clique on n nodes is denoted K_n .

In a bipartite network G , $V = P \cup S$ and $E \subseteq P \times S$ is the set of edges.

2.2.1 Comparing Clusters and Partitions

Clusters of nodes can be regarded as strong or weak. Probably the simplest definition of a strong cluster is a set of nodes which forms a clique, that is, the subgraph they induce is complete [39]. However there are less absolute ideas of community which are used more commonly.

The crucial idea behind strong and weak clusters in [43] is the degree k_i of a node i that belongs to the cluster. For a particular cluster c to which node i belongs, we separate k_i into two parts: the number of edges $k_i^{in} = \sum_{j \in c} A_{ij}$ connecting node i to other nodes in c , and the number of edges $k_i^{out} = \sum_{j \notin c} A_{ij}$ connecting node i to the nodes in the rest of the network. A strong cluster has to satisfy the condition:

$$k_i^{in} > k_i^{out}, \quad \forall i \in c \quad (2.1)$$

that is for each $i \in c$, it must have more edges to the nodes within the cluster c than edges connecting to the rest of the network. A weak cluster is defined as:

$$\sum_{i \in c} k_i^{in} > \sum_{i \in c} k_i^{out} \quad (2.2)$$

that is, the sum of all degrees for all nodes within c is larger than the sum of all degrees outgoing to the rest of the network. Obviously, a strong cluster is a weak cluster as well, but the converse is not true.

An alternative definition is proposed in [24], which relates the cluster under consideration to each other cluster and not to the whole network. Here, a strong cluster is a set of nodes where each node's degree within the cluster must be at least as large as its degree toward any other cluster in the network:

$$\forall i \in c, \quad k_i^{in} \geq \max_{c' \neq c} \left\{ \sum_{j \in c'} A_{ij} \right\}. \quad (2.3)$$

A weak cluster accordingly is one where the sum of all degrees within the cluster should be at least as large as the sum of degrees outgoing to each cluster in the network:

$$\sum_{i \in c} k_i^{in} \geq \max_{c' \neq c} \left\{ \sum_{i \in c} \sum_{j \in c'} A_{ij} \right\}. \quad (2.4)$$

Another approach, used in [18], defines a strong cluster using the betweenness centrality measurement. The betweenness is calculated for a given edge e as the number of shortest paths between every pair of nodes in the network that run through e . By iteratively removing the edges with highest betweenness centrality, components of the network will split from each other forming clusters. These hierarchies of clusters are represented in a binary tree, with nodes in a cluster more closely connected compared with other nodes in the network.

The aim of clustering algorithms is to reveal the topological structure of the network. To evaluate the communities detected by these algorithms, similarity measures have been proposed in order to assess the fit of the partition found with a desired one. A similarity measure very widely used for this purpose is Normalized Mutual Information (NMI), which tests the “goodness” of a detected partition by measuring the common information it shares with a targeted partition. The version of NMI that has been widely accepted in the literature, particularly in the LFR benchmark [27], is from [11]: given two partitions C and C' ,

$$I_{norm}(C : C') = \frac{H(C) + H(C') - H(C, C')}{(H(C) + H(C'))/2} \quad (2.5)$$

where $H(C) = -\sum_c P(c) \log P(c)$ is the Shannon entropy for partition C and $H(C, C')$ is the joint entropy between the two partitions. I_{norm} equals 1 if the two partitions are identical and 0 if they independent.

Another approach uses the Jaccard similarity coefficient for comparing two partitions of the network [8]. It is defined as the ratio of the number of node pairs classified in the same cluster in both partitions, over the number of node pairs which are classified in the same cluster in at least one partition. Let us say that a_{11} is the number of node pairs which are in the same cluster in both C and C' , a_{10} indicates the number of node pairs that are put in the same cluster in C but not in C' and a_{01} is the number of node pairs put in the same cluster in C' but not in C . The Jaccard similarity coefficient for C and C' is:

$$J(C, C') = \frac{a_{11}}{a_{11} + a_{01} + a_{10}} \quad (2.6)$$

The ratio of the Jaccard similarity coefficient for the two partitions C and C' is between 0 and 1. When $J(C, C') = 1$ the clusters in C are identical to the clusters in C' , while $J(C, C') = 0$ indicates independent clusters in both partitions, with no overlap at all.

Of course the ideal situation for measuring performance of a community detection algorithm is based on the ground truth. It requires deep knowledge of the formation of relations within and between clusters. Although it is excessively time consuming, and impractical or impossible in large networks, the result is much more accurate and more meaningful. In this chapter, we follow this approach as it provides significant outcomes.

2.2.2 Benchmarks and Performance

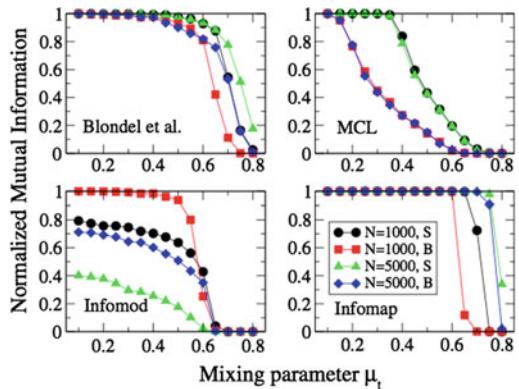
The LFR benchmark [27] allows authors of community detection algorithms to test their algorithms and evaluate the communities they have detected. It relies on creating an artificial network belonging to a “planted ℓ -partition” model, which generates a network with a given community structure. The node partitions generated in such a network can have different sizes, and different nodes can have different degrees.

This benchmark suite has the ability to test large networks of 10^3 to 10^5 nodes, to deal with overlapping communities and with both directed and undirected networks. Its novelty is that it is possible for both node degree distribution and community size to follow the power law distribution.

To run a test on the LFR benchmark, the mixing parameter μ has to be tuned at different values in the range $[0, 1]$. This mixing parameter μ is the ratio of the number of external neighbors (k^{out}) of each node by its total degree. A small value of μ indicates well separated communities, however with a high value of μ , communities overlap more and more and the community structure is weaker. This mixing parameter allows the strength of the community structure to be controlled, to be compatible with realistic topological properties. The test algorithm is run against LFR networks constructed using a selection of values for μ and the partition it finds is compared with the planted partition using NMI. The complexity of the LFR benchmark is linear in the number of edges of the constructed network, which makes performing such testing fast enough to analyse and study.

A comparative analysis of the performance of 12 community detection algorithms appears in Lancichinetti et al. [28, 29]. This study enables us to compare the stability and the accuracy of algorithms by testing them against heterogeneous distributions of node degree and community size. The outcome of this study is that the Infomap algorithm is the best algorithm to cluster very large networks, followed by the Louvain algorithm (but see the next subsection) and a Potts model algorithm. Figure 2.1 from

Fig. 2.1 The performance of four algorithms against LFR benchmark partitions. Infomap is at bottom right and Louvain (Blondel et al.) at top left [28]



[28]¹ shows the comparative performance for various μ . We describe the first two algorithms in the following subsections.

Furthermore, a recent evaluation for 11 algorithms appears in [38] where the emphasis is on the strength of community structure. It used the artificial networks generated by the LFR benchmark, where node degrees and community sizes are both power-law distributed, with a different mixing coefficient, and again the NMI is used to assess the performance of the algorithms. This evaluation concludes that Infomap is the leading algorithm on performance among all 11 algorithms.

2.2.3 Modularity Based Algorithms and the Resolution Limit

Girvan and Newman [35] initiated recent work on detecting and evaluating communities in large networks. They introduced a fast greedy technique which relies on maximising a quality function called *modularity*, defined for a partition C as

$$Q(C) = \frac{1}{2m} \sum_{ij} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta(c(i), c(j)) \quad (2.7)$$

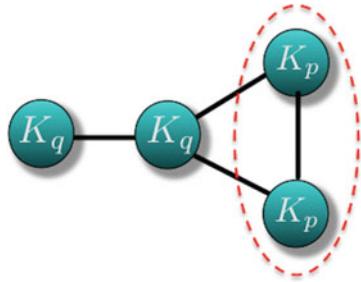
where $c(i)$ is the community to which node i is assigned, and the Kronecker delta function $\delta(c(i), c(j)) = 1$ if nodes i and j belong to the same community and 0 otherwise. The complexity of the Girvan-Newman algorithm is $O(n^3)$ and it is limited to networks with around $n = 10^3$ nodes.

Many efforts have been devoted to upgrade the computational time of modularity optimization, and extend the limit of network size that can be clustered. For instance, the Radicchi et al. [45] algorithm, in the spirit of Girvan-Newman, iteratively removes edges, but in this case removes the edges with highest clustering coefficient instead of edges with highest betweenness. The complexity of this algorithm is $O(n^2)$ which is an improvement on the greedy technique. Another example of an algorithm that takes modularity optimization as its main quality function is that of Guimera and Amaral [21].

The Walktrap algorithm proposed by Pons and Latapy [41] uses random walks to define a distance which measures the structural similarity between vertices and between communities. It is based on the idea that at some stage a random walker tends to be trapped in dense part of a network corresponding to a community. Starting from an initial assignment of each node to its own community, communities are merged according to the minimum of their distances and the process iterated. The bottom-up hierarchy is represented in a dendrogram and the algorithm stops when a partition with maximum modularity is obtained.

¹Figure reprinted with permission from Ref. [28]. ©2014 by the American Physical Society.

Fig. 2.2 Maximisation of modularity Q will fail to identify cliques in this example, e.g. if $q \gg p$, there is higher modularity for the pair of cliques K_p joined by a single edge than for the cliques themselves



However, modularity optimisation algorithms are subject to a resolution limit in the size of communities they can detect. Fortunato and Barthélémy [16] showed that communities with internal edge numbers $\leq O(\sqrt{m})$ may not be detected. Small strong communities in large networks may fail to be resolved, even when they are well defined. An illustrative example appears in Fig. 2.2. This is a definite drawback for modularity-based algorithms.

The fast modularity optimization algorithm by Blondel et al. [9], known as the *Louvain* algorithm, has one of the best results in the comparison tests [28]. The first phase of this algorithm starts by assigning each node in the network to its own community, then merging neighboring nodes that maximise value in the modularity equation. The second phase starts by dealing with previously found communities as super-nodes in a new network and repeats the first phase on this new network by merging two super nodes to achieve a higher modularity value. These steps are repeated iteratively until the maximum modularity is reached, resulting in multi-levels of communities, as super-nodes. The complexity of the Louvain algorithm is linear in the number of edges in the network, that is $O(m)$. The authors claimed the multi-level nature seems to circumvent the resolution limit problem of modularity and this appeared to be born out by its high performance evident in Fig. 2.1.

However, a very recent acknowledgement by Lancichinetti et al. [29] admits that in Fig. 2.1 they did not use the subsequent iterates of the Louvain algorithm in determining its performance, only the first phase, because the performance of the final level would be very poor, owing to the resolution limit.

2.2.4 Minimum Description Length Based Algorithms

The stochastic block model of Peixoto [40] employs minimum description length (MDL) to describe the structure of a network, through compressing the total amount of information on the network. It identifies the blocks (communities) for a network without needing to specify the number K of blocks in advance. However, there is a resolution limit in detecting the blocks which is similar to the resolution limit in modularity based algorithms: the maximum detectable block number K scales as \sqrt{n} for a fixed average degree.

The map equation method proposed by Rosvall and Bergstrom [47], known as *Infomap*, identifies communities according to information flow in the networks. Infomap has two main steps, a deterministic greedy search algorithm and then a simulated annealing approach to refine the results obtained. In its greedy search step the algorithm starts by calculating the ergodic node visit frequencies using a transition matrix to create the stationary distribution for the network.

This approach uses Huffman codes [23] to give short codewords for commonly visited nodes, and long codewords for rarely visited nodes. The quality function used to evaluate a partition is again the minimum description length MDL [19]. It measures the average length $L(C)$ in bits per step of a random walk on the network with a node partition $C = \{c_1, \dots, c_l\}$.

$$L(C) = q_{\sim} H(C) + \sum_{i=1}^l p_{\circlearrowright}^i H(P^i) \quad (2.8)$$

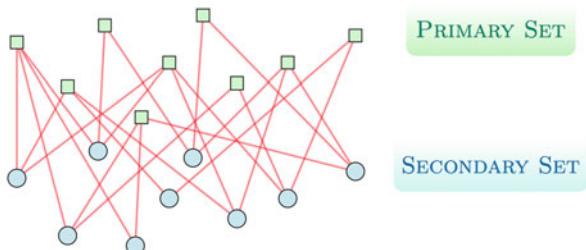
This equation has two parts: the first one is to explain the movements between the communities, where q_{\sim} is the probability that a random walker switches communities and $H(C)$ is the entropy of the community index codewords. The second part explains movements within the communities, where p_{\circlearrowright}^i is the fraction of the movements within community c_i and $H(P^i)$ is the entropy of the movements within community c_i . The complexity of the Infomap algorithm is $O(m)$.

2.3 Algorithms for Bipartite Networks

In this section we discuss community detection algorithms that are intended for bipartite networks, and the fact that the best-performed algorithm, Infomap, cannot be applied to them. Figure 2.3 illustrates the structure of a bipartite network.

Fig. 2.3 Bipartite network structure

BIPARTITE NETWORK TOPOLOGY



2.3.1 Modularity-Based Algorithms

Most authors follow the modularity method of Newman and Girvan [35] to find communities in bipartite networks. Since bipartite networks have two distinct node sets and edges only connect nodes from different sets, modularity optimization needs to be modified to identify communities in this kind of network. Guimera et al. [22] introduced a modularity measurement for bipartite networks and checked its performance against the communities in the weighted projection on P detected directly by modularity maximisation. They found no difference between these and the communities in P that resulted after projecting the communities they found in the bipartite network.

In [6], Barber developed the modularity matrix for bipartite networks, inspired by Newman's idea of a modularity matrix [34]. The modularity equation from Newman [34] takes the following form (cf. (2.7)):

$$Q = \frac{1}{2m} \sum_{ij} (A_{ij} - P_{ij}) \delta(c(i), c(j)) \quad (2.9)$$

where P_{ij} is the probability of an edge existing between i and j . Barber claims that there is a profound impact on the modularity using a normal null model in this equation, since it assigns edges at random with the expected degree of model vertices constrained to match the degrees in the actual network. Thus, he defines a null model that obeys the requirement of bipartite networks.

2.3.2 Label Propagation Algorithms

A different technique for detecting communities in unipartite networks is the Label Propagation Algorithm (LPA), proposed by Raghavan et al. [44], which uses the local network structure as a guide for finding communities in unipartite networks. LPA doesn't perform as well as Louvain and Infomap on the LFR benchmark [27]. Barber and Clark [7] introduced an extended version of LPA, denoted LPAb, for bipartite networks.

The LPAb is very fast on a bipartite network and it is an efficient method of detecting communities through maximizing the modularity optimization. Initially, it starts by assigning a unique label for each node in bipartite network that reflects their node set, so we have two different colors, say nodes in P are labelled red and in S are labelled blue. Then, nodes update their label at each step in random sequences to obtain maximization in bipartite modularity. These processes are repeated iteratively until a local maximum of bipartite modularity is reached. At the end, modules are identified as a group of nodes having the same labels. The speed of LPAb makes it the “fastest bipartite modularity optimization algorithm” [11] because the computational

time for LPAb is $O(m)$. Liu and Murata introduce an improved version of LPAb, called LPAb+ [31], which is claimed as the most reliable algorithm with highest bipartite modularity.

2.3.3 Statistical Modelling and MDL Algorithms

The MDL based stochastic block model of Peixoto [40] can be applied to a bipartite network without specifying that it is bipartite. When applied to the IMDB bipartite network (discussed in Sect. 2.5.3 below) the resulting communities from this block model fully reflect the bipartite nature, as the detectable communities partition P and S separately.

A statistical modelling approach to community detection in bipartite graphs has been proposed in [3]. The paper first surveys the statistical models used for modelling networks where actors attend events (some of these models are not intended for community detection), and of which only one (the exponential random graph or p^* model) had previously been applied to the benchmark Southern women network discussed in Sect. 2.4.2 below. It discusses a latent class model, which is a “mixed Rasch model” where the number of communities, K , is an initial (unknown) variable, and particular choices of K are fitted by assigning different event attendance probabilities among groups, but identical attendance probabilities within groups. An assumption of the model is that actors attend events independently. The choice of K is discussed at some length.

2.3.4 Infomap and the Convergence Problem

Infomap, the best performing algorithm for community detection, cannot run as intended on a bipartite network. The stationary distribution (probability of being at node i) for random walks on any network is given by the probability [30]:

$$\pi_i = \frac{k_i}{2m}, \quad i = 1, \dots, n. \quad (2.10)$$

On a bipartite network, a walk alternates between the two node sets, so, while the stationary distribution is computable, the walk doesn’t converge to it as time tends to ∞ independent of the start node. For example, if the random walk starts in one node set of a bipartite network, then it will always be in that set after an even number of steps, so the probability of being at a node in that set is zero at odd time steps. Infomap fails at its first step on a bipartite network. Thus, we can not implement Infomap on bipartite networks because of periodicity.

2.4 The Weighted Projection Approach

We cannot apply Infomap to bipartite networks directly but we can certainly apply it to a (weighted) projection onto P . Guimera et al. [22] found no difference in the node communities detected in P whether they resulted from modularity maximisation after projection, or projection after bipartite modularity maximisation. The projection method has been used for a long time in recommendation systems in the business area. Its strength is the idea that the emphasis is usually on one of the two node sets. These sets can be switched for different applications. So a weighted projection method allows us to investigate bipartite networks using powerful one mode algorithms, after a transforming process.

A projection of P in $G = (P \vee S, E)$ is a graph $G_P = (P, E_P)$ in which two nodes i and $j \in P$ are linked together if they have at least one neighbor in common in S . A projection can be weighted or unweighted but weighted projections are usually regarded as more representative of the link information in the bipartite network. Two nodes in P are more likely to have a meaningful link in reality if they have many neighbors in common, and this information should not be lost. The number of common neighbours can be represented by multiple edges between the nodes, or else by a weighted single edge between the nodes. Moreover, the information that a node in P connects to a node of degree 1 in S should not be lost.

In this section we describe the weighted projection algorithm we use, and we compare its community detection outcomes with others in the literature on a small database, which is nonetheless a benchmark for bipartite clustering techniques, the “Southern women” database [12].

2.4.1 Description and Method

Multiple edges are computationally time-consuming to process, and here we use weighted edges. Moreover, Infomap and Louvain can accept weighted networks as input.

Given G , the adjacency matrix for G_P is defined by:

$$A_{ij} = \begin{cases} 1, & \text{if nodes } i \text{ and } j \text{ have a common neighbor} \\ 1, & \text{if node } i \text{ has a neighbor which has no other} \\ & \text{neighbors in } P \text{ (resulting in a self loop at } i\text{)} \\ 0, & \text{otherwise} \end{cases}$$

For node $i \in P$, let $\Gamma(i)$ denote the set of neighbors of i ; all these are nodes in S . To measure similarity between distinct nodes i and j in P we choose the common neighbors index,

$$W_{ij} = |\Gamma(i) \cap \Gamma(j)|, \quad i \neq j \tag{2.11}$$

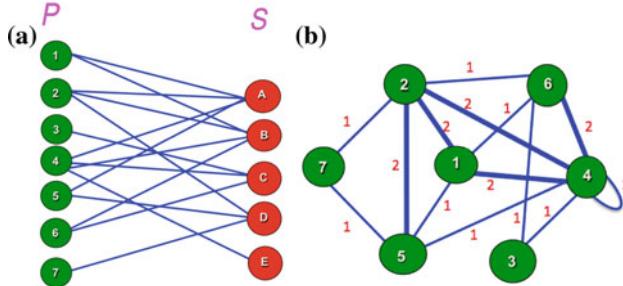


Fig. 2.4 Example of weighted projection. **a** Bipartite network with $n = 12$, $m = 15$, $|P| = 7$ and **b** weighted projection of P using (2.11) with $|E_P| = 19$

due to its simplicity and efficiency [48] on large scale networks. Then W_{ij} is the weight of the edge between i and j in the projected one mode network. This enables us to generate a weighted projected one mode network from the bipartite network in an efficient way. Further, we avoid the loss of information for a node of degree one in the secondary set S . An illustration of the weighted projection method we use appears in Fig. 2.4.

We have programmed our projection algorithm in C^{++} for compatibility with the implementations we have of the Infomap and Louvain algorithms. We start by reading the bipartite network edges as a pair of nodes, the first from P and the second from S . The labels on the nodes in this dataset do not have to be numbers, they can be post codes, book serials, bank card numbers, names of social networks or even names of people. Then, we use special techniques in C^{++} that affect the efficiency of the projection method [42]. Using a C^{++} container called Mapvector which requests a key and a value, we choose each key as an element of S and its value to be a vector of nodes in P to which it is adjacent. Then, we create pairs in a one mode network and store the result in container called “Multiset”.

To solve the labeling issue we use a mapping between strings and integers and generate new numbers that represent list of pairs with the links between nodes. After the projection we will have only (number, number) pairs which is exactly what Infomap requires. However, another issue arises here, that of losing the initial strings/labels of nodes. When we generate the final network picture we will see only links between these numbers but without any label on them. Therefore, we use the Pajek format because when declaring the nodes, we can also give a label for each node. We then input the projected network data into the Infomap² (and Louvain³) algorithms.

The pseudocode is given in Algorithm 1. We illustrate the intermediate steps of Algorithm 1 as it applies to the toy network in Fig. 2.4.

²Infomap available for download on the link: www.mapequation.org/.

³Louvain available for download on the link: <https://sites.google.com/site/findcommunities/>.

Table 2.1 Time complexity for algorithms on bipartite networks with n nodes and m edges

Algorithm	Order
LPAb	$O(m)$
LPAb+	$O(m \log^2 n)$
Algorithm 1	$O(m^2) + O(m)$

Example 2.1 In Fig. 2.4 we have $P = \{1, 2, 3, 4, 5, 6, 7\}$, $S = \{A, B, C, D, E\}$. The operation of Algorithm 1 on this network at the following lines will be:

```

4 Mapvector(string) = [(A, 1)(A, 2)(A, 4)(A, 5)(B, 1)(B, 2)(B, 4)(B, 6)
(C, 3)(C, 4)(C, 6)(D, 2)(D, 5)(D, 7)(E, 4)]
13 Mapvector[i, j] = [(A, 1 2 4 5)(B, 1 2 4 6) (C, 3 4 6) (D, 2 4 7) (E, 4)]
25 Multiset[i, j] = [(1, 2)(1, 4)(1, 5)(2, 4)(2, 5)(4, 5)
(1, 2)(1, 4)(1, 6)(2, 4)(2, 6)(4, 6)
(3, 4)(3, 6)(4, 6)
(2, 5)(2, 7)(5, 7)
(4, 4)]

```

We can compute the time complexity for the whole operation starting from converting bipartite networks to weighted unipartite networks followed by clustering them using either algorithm. The reason the projection method is also applied to the Louvain algorithm is to be able to compare the performance of Infomap with that of Louvain. The complexity for both Infomap and Louvain is $O(m)$ where m is the number of the edges in G . Our projection takes $O(m^2)$. Table 2.1 summarizes the complexity of the integrated algorithm. Although the efficiency of our algorithm is comparable with those applying bimodularity, it is not as good as those employing label propagation, as Table 2.1 shows. The running time needs to be improved.

To evaluate the quality of community detection in a bipartite network using Algorithm 1, we look to examples where it is possible to extract some ground truth. There is no suite of existing benchmark bipartite networks for testing purposes comparable to the LFR benchmarks [27] for one mode networks. The most-studied bipartite network is the very small “Southern women” network and it has been used as a de-facto benchmark for testing community detection algorithms, both for bipartite graphs (obviously not Infomap, though) and for the projection onto P .

2.4.2 Benchmark “Southern Women” Dataset

The “Southern women” network collected by Divas et al. [12] has become a benchmark for testing community detection algorithms on bipartite networks. This network has 18 women (who form the primary set P) who attended 14 different events (the secondary set S). An edge exists between two women for each event they attend

Algorithm 1: Weighted projection method for bipartite network integrated with Infomap or Louvain algorithm.

```

Require: A bipartite network.
1: initialization
2: while end of dataset not reached do
3:   read each pair from dataset
4:   store pairs in Mapvector[string]
5: end while
6: Find common neighbors:
7: for all i = 1 end of mapvector‘keys’ do
8:   print(i)
9:   for all j = 1 end of mapvector‘value’ do
10:    print(j)
11:   end for
12: end for
13: return :Mapvector[i,j]
14: Create pairs for one mode network:
15: for all i = 1 → end of Mapvector[i, j] do
16:   if size of commonneighbor = 1 “self loop” then
17:     insert the duplicate [i, i] into multiset
18:   else
19:     for i = 1 → end of Mapvector – 1 do
20:       for j = i + 1 → end of commonneighbors do
21:         insert the pair [i, j] into multiset
22:       end for
23:     end for
24:   end if
25: end for
26: return : Multiset[i, j]
27: Create the associated pairs of vertices and store them in Pajek format from this Multiset:
28: for i = 1 → end of Multiset[i, j] do
29:   store vertices in string variable ← List of vertices with its Labels
30: end for
31: while the end of Multiset not reached do
32:   currentpair = *begin of Multiset
33:   if the both pair numbers are the same then
34:     print edges[i, i]
35:     count (duplicate pairs) /* to avoid the redundant pairs */
36:   else
37:     save current pair
38:     count to list of edges string
39:     erase current pair from Multiset /* to enhance the computational time */
40:   end if
41: end while
42: store edges in string variable ← list of the edges with weights
43: Reading input network from string variable rather than from screen
44: while string variable not empty do
45:   read the input from weighted projection approach as Pajek format
46: end while
47: process the normal Infomap or Louvain algorithm
  
```

Table 2.2 The Southern women network: Number of communities of women detected by different algorithms

Algorithm	Quality function	Network applied to	Modules in P
Alzahrani et al. [5]	Modularity	Weighted projection	2
Guimera et al. [22]	Modularity	Weighted projection	2
Crampes and Plantie [10]	Bimodularity	Bipartite	3
Barber [6]	Bimodularity	Bipartite	4
Liu and Murata [31]	Bimodularity	Bipartite	4
Alzahrani et al. [5]	Map equation	Weighted projection	4

together. Most studies conducted before 2003 identify two (sometimes overlapping) communities of women while one identifies three communities [17]. In many studies, members within each community are further partitioned into core or peripheral members. More recent studies using bimodularity find more communities (3 and 4). Consequently, at least two communities are expected. In Table 2.2, we list the community numbers found in the Southern women dataset by the more recent bipartite network algorithms described in Sect. 2.3 and by our implementation of projection in Infomap and Louvain.

We compare our results for the Southern women network with results in the literature, in more detail. Using Infomap, we have community A consisting of Evelyn and Theresa (women 1 and 3, respectively), community B consisting of Katherine and Nora (women 12 and 14, respectively), and two others $C = \{8, 9, 16, 17, 18\}$ and $D = \{2, 4, 5, 6, 7, 10, 11, 13, 15\}$, as shown in Fig. 2.5. Our groups A and B consist of women frequently identified as core members of each of the two communities found in earlier studies. By contrast, Barber's two smaller communities consist of women who tended to be identified as peripheral members of each of the two communities found in earlier studies [17]. Barber also tested the success of his partition into four communities, found using the maximum bipartite modularity (as described in Sect. 2.3), as a partition in the corresponding *unweighted* projection network, and found it to have negative modularity [6]. As this is worse than considering the women as a single community, it further supports our use of the *weighted* projection network. Guimera et al. [22] found only two communities of women (red and blue) whether modularity on the unweighted projection, the weighted projection or bipartite modularity was used. They found the communities were inaccurate with unweighted projection, but identical and in agreement with supervised results in [17] for the other two methods. The total number of edges in the Southern women network after weighted projection is 139 edges. Our community A (Evelyn and Theresa) has internal edge weight 7 and lies inside the red group, while our community B (Katherine and Nora) has internal edge weight 5 and lies inside the blue group. These two “core” strong small communities are not detected by the modularity based algorithm, probably because their edge numbers fall below the resolution limit of modularity, which in this case is 12 (since $11 < \sqrt{139} < 12$). By comparison the

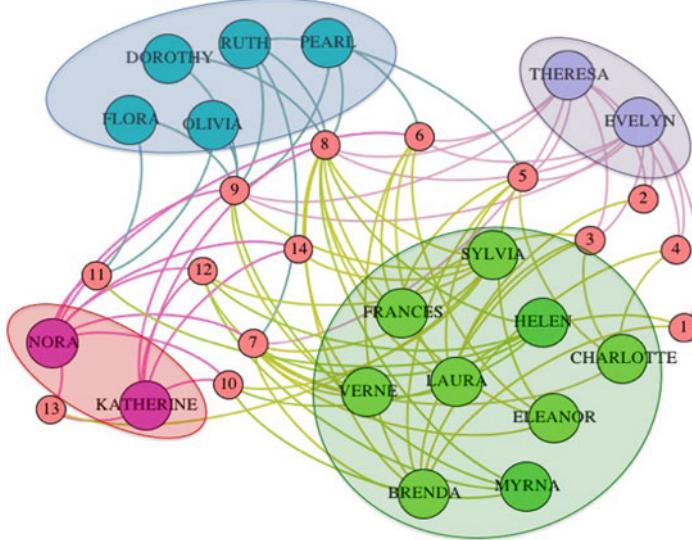


Fig. 2.5 The four communities of women found in the Southern women network. Red nodes represent S , the events the women attended, and the four other colors represent four communities within P , with nodes labelled by first name [5]

2 communities found by our projection input into Louvain have 45 and 33 internal edges. This demonstrates that, in this benchmark bipartite case, Louvain is subject to the resolution limit for modularity but Infomap is not.

2.5 Case Studies

Application of Infomap to the small and much analysed “Southern women” bipartite network shows that the communities detected represent meaningful associations between the women grouped together. In this section we continue to apply Infomap and Louvain to weighted projections of three larger bipartite networks as case studies. We demonstrate that Infomap produces meaningful communities representing some sort of ground truth, and does so better than Louvain. The case studies are presented in order of increasing size, as each highlights a different feature of Infomap community detection.

2.5.1 Noordin Top Terrorist Network

The Noordin Top terrorist group data linking individuals with relationships or affiliations first appeared in [25]. The ties or links between actors represent one or more common affiliations or relationships. Common attendance of actors at events was

Table 2.3 Communities in the Noordin Top terrorist network [4]

Algorithm	Communities	Sizes
Louvain	4	29, 16, 15, 14
Infomap	5	25, 30, 12, 4, 3

inferred from their mention together in public reports in newspapers and elsewhere. The data were coded as network data by Naval postgraduate students and the information was published in 2012 in [14]. We work with the cleaned affiliation subnetwork and thank Assoc. Prof. Murray Aitkin for providing it. It forms a bipartite network with 79 actors and 45 events (affiliations), classified into six categories (Operations, Logistics, Organizations, Training, Finance, Meeting). We excluded the actors who did not present at any of the 45 events.

In [2] the Bayesian latent class model of [3] (see Sect. 2.3.3) is applied directly to this terrorist network for $K = 1, \dots, 4$. The researchers find the $K = 3$ model fits best and use an actor’s degree to assign them to a community. Their first group consists of two important leaders and planners (Noordin Top and Azahari Husin), and they conclude that the other two groups are: the “footsoldiers”; and the intermediaries who meet the planners and train the footsoldiers.

Weighted projection of the Noordin Top bipartite network onto the actor set P determines a network with $|E_P| = 759$ edges in total weight. Using the Infomap algorithm we found 5 communities and using Louvain we found 4 communities, see Table 2.3. The modularity resolution limit for this network is $\lfloor \sqrt{759} \rfloor = 27$. Therefore, a community with strong ties and $\ll 27$ edges may not be detected by modularity based methods. The 5 communities found by the Infomap algorithm are displayed in detail in Fig. 2.6.

The smallest Louvain community (14 actors) wholly contains the third Infomap community (12 actors), and we regard them as essentially equivalent. The largest Louvain community (of 29 actors) contains 23 of the 25 actors belonging to the largest Infomap community. It also contains the smallest Infomap community (a clique of 3 actors with weighted edge sum 6). The second small Infomap community (a clique of 4 actors with weighted edge sum 6) has three actors in the largest Louvain community and one in the second largest Louvain community. *Essentially, Infomap detects three communities inside the largest Louvain community.* The two small clique communities are half an order of magnitude smaller than the modularity resolution limit. This is a real-world illustration of the phenomenon illustrated theoretically in Fig. 2.2.

Consequently, to test the communities found for meaningfulness, we concentrate on the structure found by the Infomap algorithm.

Community 4 contains actors Abdul Rauf, Imam Samudra, Apuy and Baharudin Soleh. Community 5 contains actors Enceng Kurnia, Anif Solchanudin and Salik Fridaus. These two small cliques have no recorded direct links between them, nor does Community 5 have any recorded direct links with Community 3. Identifying these small clique communities in the original bipartite network described in [46] recovers very meaningful link information. For instance, Anif Solchanudin and Salik Fridaus

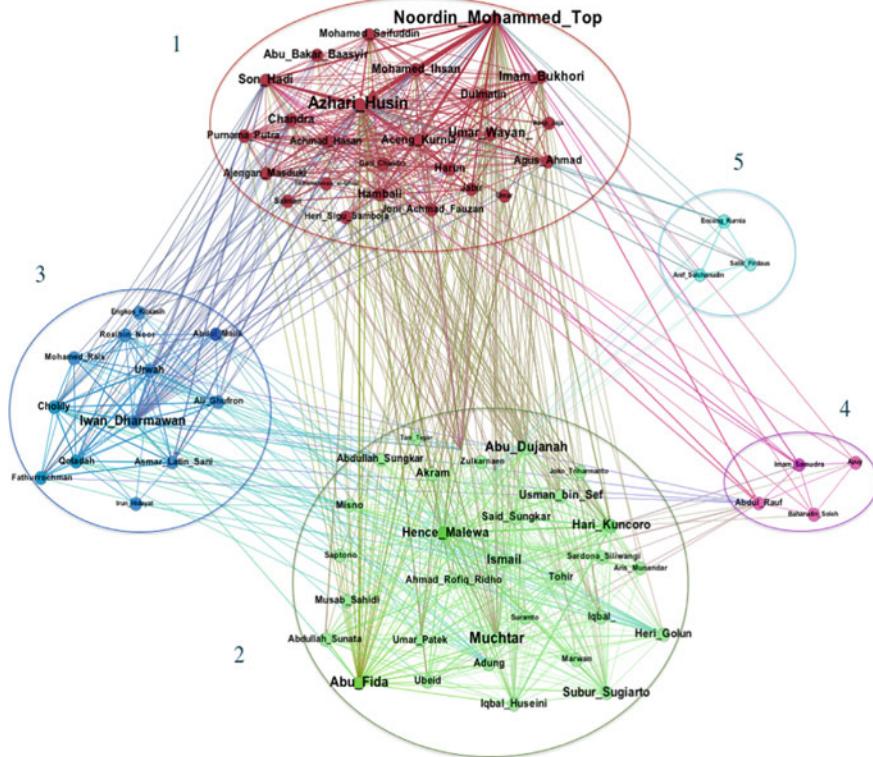


Fig. 2.6 Noordin Top terrorist network actor communities found using Infomap: Community 1 (red, top, 25 actors), Community 2 (green, bottom, 30 actors), Community 3 (purple, left, 12 actors), Community 4 (pink, below right, 4 actors), Community 5 (blue, above right, 3 actors)

were trained together to be suicide bombers for Bali Bomb II in 2005. Community 4 also reflects useful information. Abdul Rauf, Imam Samudra and Apuy came from the same organization, Ring Baten, while Apuy and Baharudin Soleh were involved directly in the Australian Embassy bombing in 2004. These two smallest communities are new structure, not found by the Louvain algorithm or in [2], and are significant from a defence analysis perspective.

In [4] we related Fig. 2.6 back to the 6 categories of events, as was done in [2] for only 3 communities, and listed the “Top Ten” actors by four different centrality measures. Community 1 contains the two principal leaders and planners (Noordin Top and Azhari Husin). In total, 8 of the 14 actors appearing in the Top Ten lists are in Community 1. The most significant common property of this group is that 17 out of 25 of its actors were affiliated to the same Organization (Jemaah Islamiyah, a transnational Southeast Asian militant Islamist terrorist organisation linked to Al-Qaeda), and we can conclude it is the most significant community.

2.5.2 NSW Crime

This publically available Australian crime data from the state of New South Wales (NSW) was published in 2012 [37]. It was collected by the NSW Bureau of Crime Statistics and Research from January 1995 to 2009, and it provides rich information about every crime that occurred in each month, categorised by offence type. There are 21 offence categories; some of these categories have subcategories that are related to the main category of the offences. For instance, “Homicide”, as a category of offence, has four subcategories (Murder, Attempted Murder, Accessory to Murder and Manslaughter) that all relate one way or another to the main category. The underlying social network of offenders is reflected in the reported crimes.

The data reports the crime according to the local government area (LGA) it was committed in. There are 155 LGAs in NSW. The bipartite network we extract has as node sets the offence categories and the LGAs that they were committed in, and has $m = 8761$. We are interested in identifying where similar patterns of crime have occurred, and which are the more dangerous areas, so P is the LGAs and S is the categories of offence. Weighted projection onto P results in an extremely dense network with $|E_P| = 3,478,084$ edges in total weight.

We applied both the Infomap and Louvain algorithms to the weighted projection on P . The Louvain algorithm did not determine any community structure at all. Consequently it is of no use for analytic purposes. However the Infomap algorithm found 2 communities of LGAs, one containing 82 LGAs and the other containing 73 LGAs. We expect there is more frequent connection between some subset of crimes for Community 1 of LGAs versus the more frequent connection between some other subset of crimes for Community 2. The modularity of this structure is higher than that for a single community, see Table 2.4, indicating it is a better structure, so the modularity-maximising Louvain algorithm should have found more than one community.

In fact it is somewhat surprising that so few communities were found. The number of internal edges in the larger community found by Infomap is 112,374, almost two orders of magnitude greater than the modularity resolution limit of $\lfloor \sqrt{3,478,084} \rfloor = 1,864$. A possible explanation is that *the communities are very weak, having a high average mixing parameter*, and so are difficult for any algorithm to detect.

However, when the LGAs in NSW are mapped and coloured according to community, a very strong geographical divide is visible. It provides a dramatic explanation of the community partition found by Infomap. Generally speaking, Community 1 includes the more populated LGAs and Community 2 includes the majority of rural and “Outback” LGAs. The 38 LGAs in the main metropolitan area, Sydney, are all in Community 1.

Table 2.4 Comparison of algorithm performance on NSW crime network [4]

Algorithm	Communities	Sizes	Modularity
Louvain	1	155	0
Infomap	2	82, 73	0.026

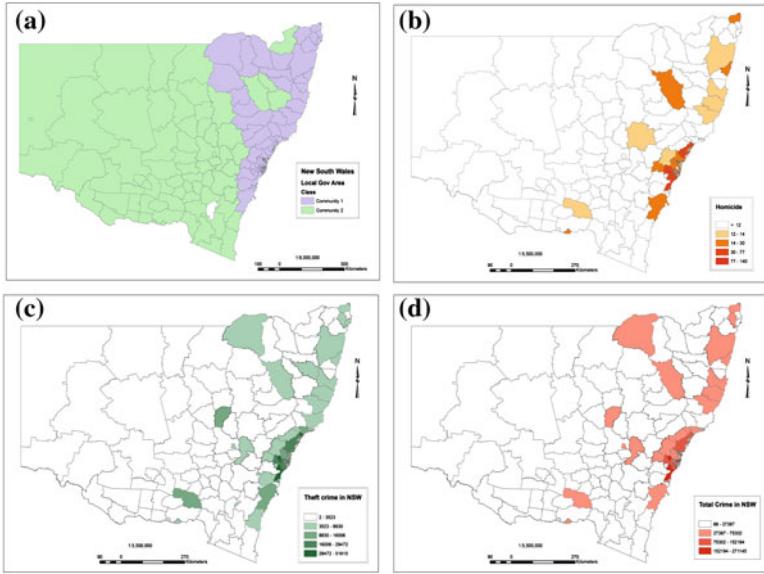


Fig. 2.7 Map of NSW local government areas and related crimes. **a** The 2 communities of LGAs found by Infomap: Community 1 contains 82 LGAs and Community 2 contains 73 LGAs. The unclassified area is the Australian Capital Territory, which is not part of NSW. Underlying crime statistics are also mapped by LGA: **b** Homicide rate; **c** Theft rate and **d** Total crime rate [4]

Analysis of the underlying crime statistics by LGA shows that for homicide (Fig. 2.7b), 90 % of the shaded LGAs occur in Community 1; for theft (Fig. 2.7c) 85 % of the shaded LGAs occur in Community 1 and for total crime rate (Fig. 2.7d), 86 % of the shaded LGAs occur in Community 1. The correlation coefficient of the crime rates between the two communities is 0.992. Deeper analysis of this network will be undertaken elsewhere.

2.5.3 Internet Movie Database Network

The Internet Movie Database (IMDB) is downloadable from [26]. We thank Dr Tiago Peixoto for the cleaned dataset from [40] that we use here. The dataset includes details about internet media and actors in them from different perspectives such as country and year of production, genre, language and rating. The “Internet Movie” term covers a range of film types, including movies, video shows, TV shows and video games; the actors are the cast members.

We are interested in the bipartite network formed from this database, where films form the primary set P and actors who have acted in a film listed in P form the secondary set S . Initially we have 275,805 actors who participated in 96,982 films. The number of edges is $\approx 1,812,697$, each edge represents an actor appearing in a film. The actors and films with degree $k \leq 1$ have been removed since they provide no

significant information on the overall network structure, giving $|P| = 96,881$. The corresponding weighted edge number is $|E_P| = 18,772,909$ and $\lfloor \sqrt{18,772,909} \rfloor = 4,332$.

The MDL stochastic block model (see Sect. 2.3.3) was applied to the whole network directly in [40] and $K = 332$ communities found, which, remarkably, perfectly reflected the underlying bipartiteness, with 165 communities entirely in P and 167 entirely in S . Note that $n = |P| + |S| = 372,787$ so $\lceil \sqrt{n} \rceil = 611$ and the maximum number of communities this algorithm can detect in the whole network is of this order.

Clustering the weighted projected network using Infomap results in 682 clusters of films in P . When we apply Louvain, only 64 clusters in P result. However, checking the four levels of the Louvain algorithm shows decreasing cluster numbers (level 0: 96,881 nodes; level 1: 528 nodes; level 2: 80 nodes; level 3: 64 nodes). In accordance with the Erratum [29], to avoid the resolution limit for modularity we take 528 as the number of film communities found by Louvain.

Thus, it seems likely that the 165 film clusters in P found by [40] is an underestimate, and the MDL stochastic block model suffers from its resolution limit in this case.

In Fig. 2.8 we plot the log degree distribution of P and the distribution of community sizes found by Infomap in P . Both demonstrate a clear heavy tail. The *community*

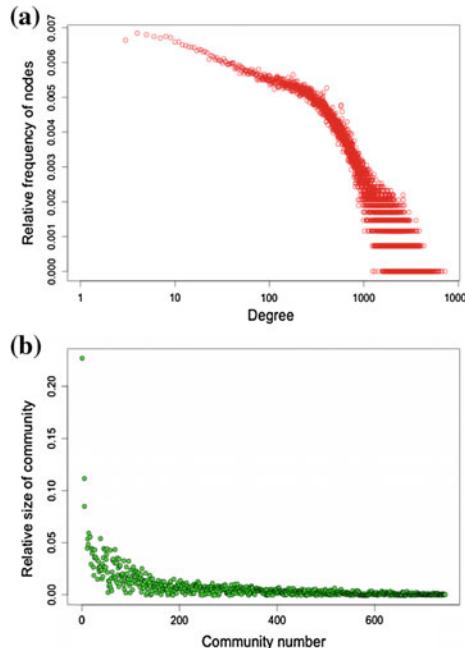


Fig. 2.8 IMDB projected film network: **a** Node degree distribution (log scale) and **b** Infomap community sizes, relative to network size

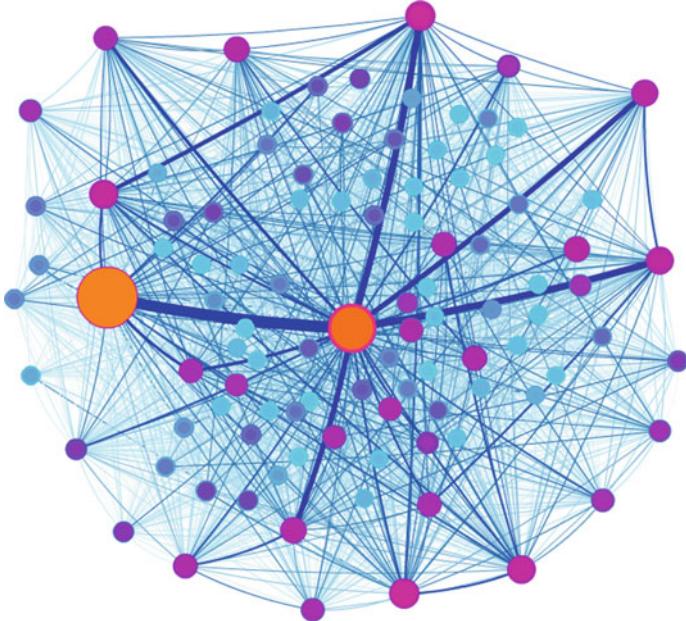


Fig. 2.9 IMDB projected film network: The largest 100 out of 682 clusters for the weighted projected network found using Infomap. It shows two giant clusters, the more central one with 10,240 nodes and the other with 22,727 nodes

size distribution is immune to any resolution limit: the smallest communities in P have 2 nodes. We conclude that in this projected network the hierarchical structure is well-defined and the communities are well separated.

For clarification, in Fig. 2.9 we show the largest 100 clusters as supernodes that clarify the structure of the projected film network. Two of the Infomap clusters are giant components; the first one has 22,727 nodes, all of which are the same film type (movie), and includes almost a quarter of P , and the second has 10,240 nodes all of which are movies as well. We checked the IMDB data briefly to see if these two clusters make sense, and they do indeed represent some ground truth. For example in the second giant cluster, almost all movies all have the same country of production (USA) and genre classification (Drama). The top 5 *hubs* (nodes with highest degree) [33] belong to these two giant components.

The second giant cluster is more central, even though it has fewer nodes than the other, for intrinsic reasons. The betweenness centrality for nodes in the second giant cluster (the number of shortest paths between node pairs in the network that pass through that node), is higher than for nodes belonging to the first giant cluster. The top three nodes for betweenness (the highest is for the 2009 movie “Never” (Part 1) and the second highest is for the 2008 movie “Around June”) and the largest hub (“Around June” with $k = 7251$) are in the second largest component.

2.6 Conclusions and Further Work

In this chapter we have reviewed a collection of community detection algorithms, including variants specifically designed for bipartite networks, that have previously been used to cluster bipartite networks. Modularity based algorithms suffer from a well-known resolution limit but the best-performing algorithm for large networks, the random-walks based Infomap, cannot be applied to a bipartite network directly.

For four bipartite networks of increasing size, we have applied Infomap to the weighted network projected onto the primary node set and compared its performance with the most popular modularity based algorithm, Louvain, and with other algorithms reported in the literature. Evaluation of detected clusters has shown that the clusters found using Infomap do embody meaningful information about the ground truth of hierarchical structure within network. Infomap can detect meaningful small communities such as cliques with sizes below the resolution limit of modularity based algorithms (the Southern women and Noordin Top terrorist networks). Infomap can detect weak large clusters better than Louvain at the upper limit of mixing coefficient (NSW crimes network). Infomap can detect a full hierarchy of clusters, that is, with no resolution limit, when they are well-defined (IMDB network).

There are number of reasons that a random walks based algorithm should be considered for community detection in bipartite networks. First, as has been our focus in this paper, it is frequently the case that the principal interest in the network is in the clustering within only one of the node sets. In this case, we believe we have shown a clear advantage in applying Infomap to detect meaningful communities in the primary projected network.

More generally, Infomap has the best performance against the LFR benchmark, so it is worthwhile to try to adapt it to bipartite networks. Moreover, the lack of existence of a benchmark for clustering algorithms on bipartite networks underlines the flexibility for researchers to employ new approaches that might suit the bipartite framework. A further reason that Infomap should be considered for bipartite networks is that it provides the sense of ground truth behind the cluster formation.

We intend to project the two sets P and S of the bipartite network in parallel, cluster them separately using the random walks based algorithm and merge their results within the bipartite network. Finally we plan to compare these bipartite communities with those clusters found by modularity-based bipartite clustering and those using multi assignment clustering.

One important observation made during the detailed study in this chapter is that nodes of the primary set might in fact belong to more than one community when the information from the secondary set is taken into account. Investigation of overlapping communities is possible future work.

Acknowledgments We are very grateful to Assoc. Prof. Murray Aitken for supplying us with the cleaned affiliation network data for the Noordin Top terrorist network; to Dr Tiago Peixoto for supplying us with the cleaned IMDB database; and to Assoc. Prof. Chris Bellman and Ms Sarah Taylor for assistance in using the ArcGIS mapping software on our clustered NSW crime network. The first author would like to thank the Ministry of Finance of Saudi Arabia for supporting his research. The work of the second author was partly supported by Department of Defence of Australia Agreement 4500743680. This work forms part of the PhD thesis of the first author, taken under the supervision of the second author.

References

1. Ahn, Y.Y., Bagrow, J.P., Lehmann, S.: Link communities reveal multiscale complexity in networks. *Nature* **466**(7307), 761–764 (2010)
2. Aitkin, M., Vu, D., Francis, B.: Statistical modelling of a terrorist network (2013)
3. Aitkin, M., Vu, D., Francis, B.: Statistical modelling of the group structure of social networks. *Soc. Netw.* **38**, 74–87 (2014)
4. Alzahrani, T., Horadam, K.J.: Analysis of two crime-related networks derived from bipartite social networks. In: Proceedings of 2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2014), pp. 890–897. IEEE (2014)
5. Alzahrani, T., Horadam, K.J., Boztas, S.: Community detection in bipartite networks using random walks. *Proceedings of CompleNet 2014. Springer Studies in Computational Intelligence*, vol. 549, pp. 157–165 (2014)
6. Barber, M.J.: Modularity and community detection in bipartite networks. *Phys. Rev. E* **76**(6), 066102 (2007)
7. Barber, M.J., Clark, J.W.: Detecting network communities by propagating labels under constraints. *Phys. Rev. E* **80**(2), 026129 (2009)
8. Ben-Hur, A., Elisseeff, A., Guyon, I.: A stability based method for discovering structure in clustered data. *Pac. Symp. Biocomput.* **7**, 6–17 (2002)
9. Blondel, V.D., Guillaume, J.L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. *J. Stat. Mech.: Theory Exp.* **2008**, P10008 (2008)
10. Crampes, M., Plantie, M.: A unified community detection, visualization and analysis method (2013). arXiv preprint [arXiv:1301.7006](https://arxiv.org/abs/1301.7006)
11. Danon, L., Diaz-Guilera, A., Duch, J., Arenas, A.: Comparing community structure identification. *J. Stat. Mech.: Theory Exp.* **2005**, P09008 (2005)
12. Davis, A., Gardner, B.B., Gardner, M.R.: Deep South: A Social Anthropological Study of Caste and Class. University of Chicago Press, Chicago (1941)
13. Evans, T., Lambiotte, R.: Line graphs, link partitions, and overlapping communities. *Phys. Rev. E* **80**(1), 016105 (2009)
14. Everton, S.F.: Disrupting Dark Networks. Cambridge University Press, Cambridge (2012)
15. Fortunato, S.: Community detection in graphs. *Phys. Rep.* **486**(3), 75–174 (2010)
16. Fortunato, S., Barthelemy, M.: Resolution limit in community detection. *Proc. Natl. Acad. Sci. USA* **104**(1), 36–41 (2007)
17. Freeman, L.C.: Finding social groups: a meta-analysis of the southern women data. In: Breiger, R., Carley, K.M., Pattison, P. (eds.) *Dynamic Social Network Modeling and Analysis*, pp. 39–97. National Academies Press, Washington (2003)
18. Girvan, M., Newman, M.E.: Community structure in social and biological networks. *Proc. Natl. Acad. Sci. USA* **99**(12), 7821–7826 (2002)
19. Grunwald, P.D., Myung, I.J., Pitt, M.A.: *Advances in Minimum Description Length: Theory and Applications*. MIT Press, Cambridge (2005)
20. Guillaume, J.L., Latapy, M.: Bipartite structure of all complex networks. *Inf. Process. Lett.* **90**(5), 215–221 (2004)

21. Guimera, R., Sales-Pardo, M., Amaral, L.S.A.N.: Modularity from fluctuations in random graphs and complex networks. *Phys. Rev. E* **70**(2), 025101 (2004)
22. Guimera, R., Sales-Pardo, M., Amaral, L.S.A.N.: Module identification in bipartite and directed networks. *Phys. Rev. E* **76**(3), 036102 (2007)
23. Huffman, D.A.: A method for the construction of minimum-redundancy codes. *Proc. IRE* **40**(9), 1098–1101 (1952)
24. Hu, Y., Chen, H., Zhang, P., Li, M., Di, Z., Fan, Y.: Comparative definition of community and corresponding identifying algorithm. *Phys. Rev. E* **78**(2), 026121 (2008)
25. International Crisis Group: Terrorism in Indonesia: Noordin's Networks. Asia Report no. 114, Brussels, Belgium (2006)
26. Internet Movie Database original database [Online]
27. Lancichinetti, A., Fortunato, S., Radicchi, F.: Benchmark graphs for testing community detection algorithms. *Phys. Rev. E* **78**(4), 046110 (2008)
28. Lancichinetti, A., Fortunato, S.: Community detection algorithms: a comparative analysis. *Phys. Rev. E* **80**(5), 056117 (2009)
29. Lancichinetti, A., Fortunato, S.: Erratum: Community detection algorithms: A comparative analysis. *Phys. Rev. E* **80**, 056117 (2009) (*Phys. Rev. E*, **89**(4), 049902 (2014))
30. Levin, D.A., Peres, Y., Wilmer, E.L.: Markov Chains and Mixing Times. American Mathematical Society, Providence (2009)
31. Liu, X., Murata, T.: An efficient algorithm for optimizing bipartite modularity in bipartite networks. *JACIII* **14**, 408–415 (2010)
32. Mukherjee, A., Choudhury, M., Ganguly, N.: Understanding how both the partition of a bipartite network affect its one-mode projection. *Phys. A* **390**(20), 3602–3607 (2011)
33. Newman, M.E.: The structure and function of complex networks. *SIAM Rev.* **45**(2), 167–256 (2003)
34. Newman, M.E.: Finding community structure in networks using the eigenvectors of matrices. *Phys. Rev. E* **74**(3), 036104 (2006)
35. Newman, M.E., Girvan, M.: Finding and evaluating community structure in networks. *Phys. Rev. E* **69**(2), 026113 (2004)
36. Nishikawa, T., Motter, A.E., Lai, Y.C., Hoppensteadt, F.C.: Heterogeneity in oscillator networks: are smaller worlds easier to synchronize? *Phys. Rev. Lett.* **91**(1), 014101 (2003)
37. NSW Bureau of Crime Statistics and Research. Dataset [Online]. NSW Crime data (2008)
38. Orman, G.K., Labatut, V., Cherifi, H.: On accuracy of community structure discovery algorithms (2011). arXiv preprint [arXiv:1112.4134](https://arxiv.org/abs/1112.4134)
39. Palla, G., Derenyi, I., Farkas, I., Vicsek, T.: Uncovering the overlapping community structure of complex networks in nature and society. *Nature* **435**(7043), 814–818 (2005)
40. Peixoto, T.P.: Parsimonious module inference in large networks. *Phys. Rev. Lett.* **110**, 148701 (2013). (Erratum. *Phys. Rev. Lett.* **110**(16), 169905 (2013))
41. Pons, P., Latapy, M.: Computing communities in large networks using random walks. *J. Graph Algorithms Appl.* **10**(2), 191–218 (2006)
42. Preiss, B.R.: Data Structures and Algorithms with Object-Oriented Design Patterns in C++. Wiley Press, New York (1997)
43. Radicchi, F., Castellano, C., Cecconi, F., Loreto, V., Parisi, D.: Defining and identifying communities in networks. *Proc. Natl. Acad. Sci. USA* **101**(9), 2658–2663 (2004)
44. Raghavan, U.N., Albert, R.K., Kumara, S.: Near linear time algorithm to detect community structures in large-scale networks. *Phys. Rev. E* **76**(3), 036106 (2007)
45. Radicchi, F., Castellano, C., Cecconi, F., Loreto, V., Parisi, D.: Defining and identifying communities in networks. *Proc. Natl. Acad. Sci.* **101**(9), 2658–2663 (2004)
46. Roberts, N., Everton, S.F.: Strategies for combating dark networks. *J. Soc. Struct.* **12**, 1–32 (2011)
47. Rosvall, M., Bergstrom, C.T.: Maps of random walks on complex networks reveal community structure. *Proc. Natl. Acad. Sci.* **105**(4), 1118–1123 (2008)
48. Zhou, T., Lu, L., Zhang, Y.C.: Predicting missing links via local information. *Eur. Phys. J. B* **71**(4), 623–630 (2009)

Chapter 3

Epidemiological Modeling on Complex Networks

Zhen Jin, Shuping Li, Xiaoguang Zhang, Juping Zhang
and Xiao-Long Peng

Abstract The present chapter is devoted to review some literatures on the modeling infectious disease on complex networks. From the following several aspects we give a brief summary about solving the problem of the disease spread: Modeling approaches of epidemic dynamics on complex networks, Application of percolation theory in propagation dynamics, Epidemic models in complex network with demographics and Epidemic spreading on multilayer networks. In the first section, the Node-based and Edge-based mean-field modeling approaches on complex networks are reviewed and compared respectively, and the second section reviews the application of bond percolation in the single network (undirected graphs, directed graphs, bipartite graphs and clustered networks) and coupled networks (overlap networks and interconnected networks), then gives a review about the disease epidemics and site or bond percolation or both site and bond percolation in small-world networks. Following, we present an overview on some of recent studies on epidemic dynamics with demographics and epidemic processes on multilayer networks in the last two section, respectively.

Z. Jin (✉) · X. Zhang · X.-L. Peng
Complex Systems Research Center, Shanxi University, Taiyuan 030006,
People's Republic of China
e-mail: jinzhn@263.net

X. Zhang
e-mail: zhxg0320@sxu.edu.cn

X.-L. Peng
e-mail: xlpeng@sxu.edu.cn

S. Li · J. Zhang
Department of Mathematics, North University of China, Taiyuan 030051,
People's Republic of China
e-mail: lspnuc@126.com

J. Zhang
e-mail: zhangjuping@foxmail.com

3.1 Epidemiological Processes Taking Place on Complex Networks

Since the first mathematical method used to research the spread of disease by Daniel Bernoulli in 1760, mathematical models of infectious disease spread have played a significant role in improving our understanding of epidemics and developing better intervention strategies [1]. The representative work of Kermack and McKendrick giving the definition of compartment models of infectious diseases, which is also the main modeling methods to epidemic. The compartment models generally assume that the population can be divided into different classes or compartments depending on the stage of the disease, such as susceptibles (denoted by S, those who can contract the infection), infectious (I, those who contracted the infection and are contagious), and recovered (R, those who recovered from the disease). The early epidemic models assume that a population of individuals is mixed homogeneously, i.e. each pair of individuals have the same contact probability. This framework assume the interactive individuals are mixed homogeneously, the epidemic compartment model neglects the significance of population connectivity. Such simplification can hardly solve new puzzles emerged in the present networking society. Hence, the epidemic models based on networks are developed to eliminate some of the standard homogeneous assumptions of mathematical models.

In very general terms a network is any system that admits an abstract mathematical representation as a graph whose nodes denoted the different individuals and in which the links represent the interaction among these individuals. With individuals represented as nodes, and infection of one by another as a link, any epidemic model can be thought of as the spreading process on network. The network theory have provided a new modelling approach for population epidemic dynamics. In recent years network models have been notably successful in analysing models where individuals vary greatly in their number of contacts (the degree distribution of the underlying graph). And several theoretical approaches for epidemic modelling on networks have been developed. From the view of modeling approaches, which can be divided by the node-based mean-field approaches and edge-based mean-field approaches, we review these approaches in the following.

3.1.1 Node-Based Mean-Field Approaches

Node-based mean-field theory was the first theoretical approach proposed for the analysis of epidemic spreading processes on complex networks [2–6]. This method not only divide the population in terms of the state of individuals, but also as the degree of the nodes. The basic assumption is all of the nodes with the same degree have the same statistically properties, so the nodes with the same degree can be divided into the same class. And any given node with degree k is connected with

the same probability to any node with degree l , which denoted as $p(l|k)$ [4, 5]. This method is convenient to take into account the topology structure of the network by a simple way.

Node-based mean-field epidemic models focuses on the relative densities of the individuals of degree k in the special class, such as susceptible (s_k), or infective compartment (ρ_k),

$$s_k = \frac{S_k}{N_k}, \quad \rho_k = \frac{I_k}{N_k}, \quad (3.1)$$

i.e. the probability that an individual in state S or I with degree k in the susceptible or infective compartment. The SIS model is described in terms of the relative densities infective nodes, in which the infection rate of a susceptible is proportional to its degree and the probability of a random edge connected by infective node [2, 3]. The SIS dynamical equation for ρ_k is derived by

$$\frac{d\rho_k(t)}{dt} = -\rho_k(t) + \lambda k[1 - \rho_k(t)] \sum_{lp} (l|k) \rho_l(t), \quad (3.2)$$

in which λ represents the effective infection rate.

For the set Eq.(3.2), Boguñá and Pastor-Satorras gave an analysis to derive the epidemic threshold, which is

$$\lambda_c = \frac{1}{\Lambda_m}, \quad (3.3)$$

where Λ_m is the largest eigenvalue of the connectivity matrix C [4], whose elements are $C_{kl} = kp(l|k)$. And the paper [3] give the specific form of the epidemic threshold in the case of uncorrelated networks, in which $p(l|k) = lp(l)/\langle k \rangle$, and $\langle k \rangle$ is the average degree of the network. Thus in uncorrelated networks $C_{kl} = klp(l)/\langle k \rangle$, and the general expression of epidemic threshold on uncorrelated networks, which is

$$\lambda_c = \frac{\langle k \rangle}{\langle k^2 \rangle}. \quad (3.4)$$

It is needed to note that for particular choices of the degree $p(k)$, there have different forms of epidemic threshold. For example, for a power-law degree distribution of the form, Pastor-Satorras and Vespignani solve it by making an integral approximation, and thus show that there is no nonzero epidemic threshold for the SIS model in the scale-free networks [2, 7, 8].

The analysis above is the general case of the epidemic model on networks, especially for the epidemic spreading on heterogeneous networks. Since $\langle k^2 \rangle = \langle k \rangle$ holds for an homogeneous network, so the generation expression of epidemic threshold (3.4) recovers the result $\lambda_c = 1/\langle k \rangle$ in homogeneous network. Furthermore, Wang et al. gave a strict global analysis of the SIS model (3.2), which can be used to prove the conclusion obtained above [9].

This mean-field method can also be applied to the SIR transmission type model [10], which can derive the expected final size during the whole transmission process. In addition, this modeling method has been also generalized to a number of other cases, including other degree distributions [3], Immunization of complex networks [6], other type of networks [11–15]. In the latter case, Meyers et al. made use of the generating function methodology to study the spreading probabilities of epidemic on asymmetry networks [12], Wang et al. established the dynamic of mean-field equations on directed networks, and obtaining the epidemic outbreak threshold analytically [11], and Zhang et al. developed an epidemic model for an SIS infection based on semi-directed networks to capture the coexist of directed and undirected contacts, more general global analysis are given, the solution contains the results on undirected networks and directed networks as special cases [13]. Zhang Juping et al. studied the spreading of sexually transmitted diseases on bipartite scale-free graphs, representing heterosexual and homosexual contact networks [16]. And Wang et al. extended this method to take into account the effect of vectors, such as mosquitoes, and the reproduction number is obtained, which can be used to examine the effect of vectors on the epidemic spreading [17]. Kiss et al. specified SIR type transmission both through global homogeneous mixing and via a contact network, extended the model to incorporate the homogeneous as well as heterogeneous contacts simultaneous [18]. Of particular note is that Ma Junling et al. extended this method to effective degree network disease models [19], modified the deficiencies the SIS network epidemic model: an infectious individual with one transmittable link can transmit to multiple susceptibles before it recovers, so the effective degree model deal with the problem of Pastor-Satorras and Vespignani SIS model which overestimates the epidemic transmission threshold. Besides, relative to the static network epidemic model, Jin et al. established a node-based epidemic model on dynamic networks, which can study the effect of demographics on disease transmission [20].

3.1.2 Edge-Based Mean-Field Approaches

Comparing to the node-based epidemic model, in order to understand how contact networks influence disease spread and the effect of the edge on the disease spread, edge-based mean-field epidemic models are proposed. In this subsection we discuss their advantages and disadvantages for different edge-based infectious disease models, and their differences with node-based epidemic model.

Pairwise Model. An alternative approach, aimed to describing the effect of edges, is pairwise epidemic model. The pairwise models, based on pair approximation methods, study that the rate susceptible individuals become infected is proportional to the number of partnerships between susceptible and infected individuals, and the purpose is to study how the number of individuals and kinds of partnerships changes in time [21, 22]. The expected number of susceptible-susceptible and susceptible-infected partnerships in the population can be denoted as $[SI]$, $[SS]$, the other type of partnerships can be defined in the same way. Then the rate at which susceptible

individuals become infected clearly depends on the number of susceptible-infected partnerships, which can be represented by $[SI]$ in the view of mean-field model. And there exists flux between $[SI]$ and $[SS]$ due to the recover of infected individuals (SIS model), and susceptible individuals becoming infected from another partnership.

Keeling et al. established the SEIRS pairwise model to research the childhood epidemic [23]. Later Keeling proposed a SIR pairwise model on homogenous networks, this method can provide a more general framework and neighbourhood structure, which can be also used to describe the behavior of spatial models in terms of ordinary differential equations [21]. The proposed pairwise model is

$$[\dot{S}] = -\tau[SI], \quad (3.5)$$

$$[\dot{I}] = \tau[SI] - \gamma[I], \quad (3.6)$$

$$[\dot{SS}] = -2\tau[SSI], \quad (3.7)$$

$$[\dot{SI}] = \tau([SSI] - [ISI] - [SI]) - \gamma[SI], \quad (3.8)$$

$$[\dot{II}] = 2\tau([ISI] + [SI]) - 2\gamma[II]. \quad (3.9)$$

where τ and γ represent the infection rate and recovery rate, respectively. And they proposed the pair approximation methods on homogeneous networks

$$[ABC] \approx \frac{\langle k \rangle - 1}{\langle k \rangle} \frac{[AB][BC]}{[B]} \left(1 - \phi + \phi \frac{N}{\langle k \rangle} \frac{[AC]}{[A][C]} \right), \quad (3.10)$$

and ϕ is the clustering coefficient of the network. Following, Sharkey et al. established a pair-level differential equation model for SIR infection model based on asymmetric contact networks, which extended the results of pairwise model on symmetric contact networks to asymmetric contact networks [24]. Eames established a pairwise model to explore the difference between regular and random contacts, considering particularly the effect of clustering within the contact network [25]. Taylor et al. formalised the link between the SIS pairwise model on homogeneous networks and the exact Markovian formulation, involving the rigorous derivation of the exact ODE model at the level of pairs in terms of the expected number of pairs and triples. And a new interpretation of both closures is presented, which explains several of their previously observed properties [26]. Furthermore, Gross first develop the pairwise model on adaptive networks to investigate the effect of adaptive topology for simple models of infectious spreading of SIS dynamics [27], in which the possibility for individuals to protect themselves by avoiding contacts with infected people is considered. The results show that the adaptive topology can lead to rich dynamics, such as the bistability, saddle-node bifurcation and Hopf bifurcation.

Besides, Eames and Keeling also established the SIS pairwise model on heterogeneous networks to study the spreading of the sexually transmitted diseases, developing an intuitive mathematical framework to deal with the heterogeneities implicit within contact networks and those that arise because of the infection process [28, 29], they also proposed the pair approximation methods on heterogeneous networks

$$[A_k B_l C_m] \approx \frac{l-1}{l} \left((1-\phi) \frac{[A_k B_l][B_l C_m]}{[B_l]} + \phi \frac{\langle k \rangle N}{km} \frac{[A_k B_l][B_l C_m][C_m A_k]}{[A_k][B_l][C_m]} \right). \quad (3.11)$$

Based on the pairwise epidemic model on heterogeneous networks, House et al. proposed several kinds of approximation methods for the pairwise model, and analysed that heterogeneous-mixing node-based models could be derived from a general pairwise approach [30]. Furthermore, Simon et al. presented a new closure relation that involved not only the average degree but also the second and third moments of the degree distribution, then they developed a super SIS compact pairwise model on heterogeneous networks, showing that the new approximate method agreed well with heterogeneous pairwise models which are consist of significant differential equation [31].

Edge-based compartment model. Based on the probability generating function for the degree distribution, Volz introduced a dynamical edge-based epidemic model, which has the advantage of using only three differential equations [32]. This approach dynamically describes the SIR disease process, while accounting for the network structure. The model incorporates the probabilities that a neighbor of a susceptible individual is susceptible or infectious, which can be denotes as

$$P_I = \frac{[SI]}{\sum_k kS_k}, \quad P_S = \frac{[SS]}{\sum_k kS_k}, \quad (3.12)$$

respectively. $\theta(t)$ represents the fraction of degree one nodes that remain susceptible at time t , thus if a node has k contacts, its probability of being susceptible is θ^k , and $G_0(x) = \sum_k p(k)x^k$ denotes the probability generating function of the degree distribution $p(k)$, then the edge-based compartment model proposed by Volz is

$$\dot{\theta} = -\tau P_I \theta, \quad (3.13)$$

$$\dot{P}_I = -(\tau + \gamma)P_I + \tau \theta \frac{G_0''(\theta)}{G_0'(\theta)} P_S P_I + \tau P_I^2, \quad (3.14)$$

$$\dot{P}_S = -\tau \theta \frac{G_0''(\theta)}{G_0'(\theta)} P_S P_I + \tau P_S P_I, \quad (3.15)$$

$$S = NG_0(\theta), \quad (3.16)$$

$$I = \tau P_I N \theta G_0'(\theta) - \gamma I. \quad (3.17)$$

which shows that it is possible to investigate the dynamics of epidemic spread on complex networks using a coupled system of only three ODEs (the first three equations above).

Furthermore, Miller derived a single differential equation with only a single higher order term that governs the dynamics [33, 34], which is

$$\dot{\theta} = -\tau\theta + \tau \frac{G'_0(\theta)}{G'_0(1)} + \gamma(1 - \theta), \quad (3.18)$$

$$S = NG_0(\theta), \quad (3.19)$$

$$I = N - S - R, \quad (3.20)$$

$$\dot{R} = \gamma I. \quad (3.21)$$

This framework can develop to calculate the dynamics, and can also be applied to predict the final size of an epidemic in a concise way. The equations derived by Miller are simpler, and the terms in the equations are more easily interpreted. The resulting calculations for the numbers of susceptible, infected, and recovered individuals are of comparable complexity to the standard mass-action SIR equations, but allow for more realistic population interactions [33, 34]. It is shown that calculations for both the final size and the dynamics of an epidemic on a random network can be placed into a common framework. Again, the Edge-based modeling approach is also used to study the epidemic spreading on dynamic networks [34].

The form of the Edge-based epidemic model is simpler than the other model, and Ma junling have shown that the Edge-based epidemic model appears to best approximate with the stochastic SIR epidemic process on a contact network [19], i.e. the Edge-based epidemic model is more accurate than node-based epidemic model. However, the Edge-based epidemic model cannot be easily extended to SIS type diseases. Therefore, identifying appropriate closures which satisfy the dynamics in modeling SIS disease are still the challenges for the future study.

3.2 Application of Percolation Theory in Propagation Dynamics

The structure of networks and the function of networks interact with each other. On the one hand, the structure of complex networks can influence the function of networks. For example, the contacts structure among individuals must influence the transmission of the disease when the disease is transmitted on social networks in real lives. In turn, if individuals take immune vaccinations of diseases, it must destroy paths and influence the connectivity of the network to prevent those individuals from catching the disease. Thus the function of complex networks can influence the structure of networks. In the above example, it is obvious that the extent of influence is decided by the resilience of the network. One of measure methods, for the resilience, is the variation in the fraction of vertices in the largest connected subgraph of the network, which is equal to the giant connected component. In addition, the propagation process on the network and percolation on the network are very similar, the relation between them is shown as Fig. 3.1.

In sum, the percolation model can be used to study the spread of disease on the network.

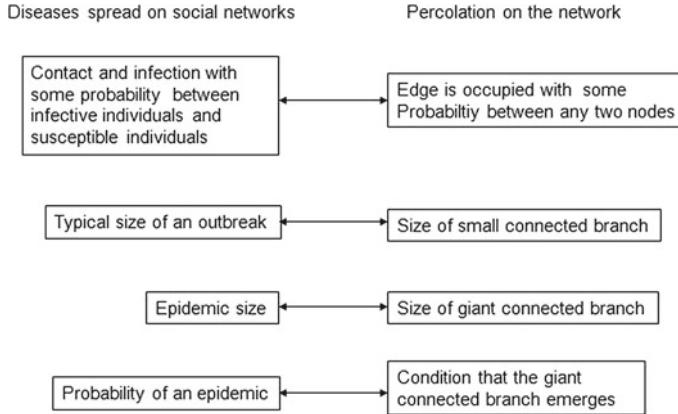


Fig. 3.1 Relation between the transmission of the disease and the percolation

3.2.1 Bond Percolation

Diseases spreading in the single network. Here, it is assumed the disease starts with a single infective individual in the network, each edge in the network where the transmission happens is marked or called “occupied” with some probability. The transmission process of the disease on the network is studied and equivalent to a bond percolation with some occupation probability of edges. The outbreak size of the disease, the epidemic size and the threshold of an epidemic can be discussed by the bond percolation process on the network. Based on the hypothesis that the network has the tree or tree-like structure, Newman et al. study the transmission of diseases in the different types of networks by the theory of bond percolation. Next, we summarize some applications of the theory of bond percolation to an SIR dynamical process.

Newman studied an SIR model which describes the spread of disease on the social network by the method of bond percolation in 2002 [7]. In this reference, the traditional SIR model is modified as follows: (1) the number of contacts with others of each individual is not same, and it is following some distribution, namely, the network where the disease is transmitted has some degree distribution. (2) the probability of disease-causing contact between pairs of individuals is also not same. The average rate of disease-causing contacts between susceptible individual j and infective individual i is denoted as r_{ij} , and the time that the infective individual has infectivity is denoted as τ_i . The probability of transmission T_{ij} between susceptible individual j and infective individual i can be obtained,

$$T_{ij} = 1 - e^{-r_{ij}\tau_i}. \quad (3.22)$$

In this paper, at first, r_{ij} and τ_i are assumed to be two iid random variables. Once distributions of r_{ij} and τ_i are given, the average T of T_{ij} over these two distribution is

obtained. The transmission process of the disease on random graphs with arbitrary degree distributions p_k is studied and equivalent to a problem of bond percolation with the occupation probability T . $G_0(x)$ represents the probability generation function of the degree distribution, then

$$G_0(x) = \sum_{k=0}^{\infty} p_k x^k. \quad (3.23)$$

Furthermore, the probability generation function of the excess degree distribution of nodes can be obtained $G_1(x) = \frac{G'_0(x)}{G'_0(1)}$. The average outbreak size of diseases is

$$\langle s \rangle = 1 + \frac{T G'_0(1)}{1 - T G'_1(1)}. \quad (3.24)$$

It is obvious that the critical transmissibility T_c is

$$T_c = \frac{1}{G'_1(1)} = \frac{\Sigma_k k p_k}{\Sigma_k k(k-1) p_k}. \quad (3.25)$$

Formula (3.25) implies that the critical transmissibility T_c is the reciprocal of the excess average degree. Namely the larger the excess average degree, the smaller the critical transmissibility T_c , then the disease is more easy to be epidemic. Above the epidemic threshold, the epidemic size is

$$S(T) = 1 - G_0(u; T), \quad (3.26)$$

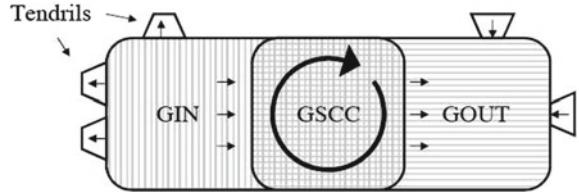
where $u = G_1(u; T)$, $G_0(x, T) = G_0(1 + (x-1)T)$, $G_1(x, T) = G_1(1 + (x-1)T)$. When infectiveness times and transmission probabilities are nonuniform and correlated, the disease spread is also discussed.

It is noted that the above network is a undirected graph. Newman expanded the percolation theory in random directed graphs [35]. It is different from undirect graph that there are three giant connect components shown as Fig. 3.2, the giant strongly connected component (GSCC), the giant out-component (GOUT) and the giant in-component (GIN) are presented respectively. The condition that three giant connect components happen is same, and, the explicit condition is

$$\sum_{jk} (2jk - j - k) p_{jk} = 0. \quad (3.27)$$

On the basis of the network in Ref. [35], using the theory of bond percolation, Meyers et al. develop a mathematical framework to predict disease transmission through semi-directed contact networks [12]. In that network, some edges are undirected, which represent that the probability of transmission is symmetric between individuals located at both ends the edge, while other edges are directed. The probabilities

Fig. 3.2 Structure of a semi-directed network [12]



that the disease is transmitted in different directions are different. The expected size of the outbreak and the probability of an epidemic are given. Sizes of three giant connect components reached by occupied directed and undirected edges are given respectively. Authors applied these methods to control severe respiratory diseases, and obtained that health care workers are vulnerable and the hospital-based containment during outbreaks is very important.

In addition, the transmission of disease on a structured network, especially the spread of a sexually transmitted disease on a bipartite graph [7], is studied. The disease is only transmitted between male and female. T_{fm} and T_{mf} represent the transmission probability from female to male and from male to female respectively. In this bipartite network, the degree distribution of male (female) nodes is $p_j(q_k)$, the average degree of male (female) nodes is $\mu(v)$. Generating functions of degree distributions and excess degree distributions about males and females are given,

$$f_0(x) = \sum_j p_j x^j, \quad f_1(x) = \frac{1}{\mu} f'_0(x), \quad (3.28)$$

$$g_0(x) = \sum_k q_k x^k, \quad g_1(x) = \frac{1}{v} g'_0(x). \quad (3.29)$$

When there is no epidemic, the average outbreak size for male and female is respectively

$$\langle s \rangle_m = 1 + \frac{T_{mf} T_{fm} f'_0(1) g'_1(1)}{1 - T_{mf} T_{fm} f'_1(1) g'_1(1)}, \quad (3.30)$$

and

$$\langle s \rangle_f = 1 + \frac{T_{mf} T_{fm} g'_0(1) f'_1(1)}{1 - T_{mf} T_{fm} f'_1(1) g'_1(1)}, \quad (3.31)$$

the condition of phase transition is

$$T_{mf} T_{fm} f'_1(1) g'_1(1) = 1. \quad (3.32)$$

Considering there exists a large number of community in real networks, Newman find that clustering increases the basic reproduction number to make the disease is

more easy to epidemic while clustering significantly reduces the size of epidemics once the average degree of networks is fixed by the theory of percolation [36]. The result is contrary to that the cluster decreases the basic reproduction obtained by the theory bond percolation [37–43].

Diseases spreading in the coupled network. In real lives, there are some coupled networks that are coupled of several single networks, in those single networks, nodes have different types or edges have different properties, and all single networks have different topology structures. There are several types of coupled network: multi-layer network, interdependent network, interconnected network or interacting network, multidimensional network, multilevel network, hypergraph [44]. Next, we will introduce applications of the theory of bond percolation in the coupled network.

In 2005, Newman investigated that two pathogens compete in a single network which is thought as a interdependent network [45]. In this paper, it is assumed that the degree distribution of the network is p_k . The author firstly consider the transmission of the first pathogen in the network. It is obvious that the critical value of the transmission probability T_c is $\frac{1}{G'_1(1)}$, and the size of the giant connected component is S when the first disease is epidemic (see the undirected network). Subsequently the second pathogen is transmitted in the residual graph with the size $1 - S$. For a node with degree k , $P(\text{uninf.}, m|k)$ is the probability that it isn't infected after the first pathogen is epidemic and has m neighbor nodes uninfected by the second pathogen. The degree distribution of nodes in the residual graph is $P(m|\text{uninf.}) = \frac{1}{G_0(u)} \sum_{k=m}^{\infty} p_k C_k^m [G_1(\tilde{u})]^m [\tilde{u} - G_1(\tilde{u})]^{k-m}$, where \tilde{u} is the solution of the equation $\tilde{u} = 1 - T + TG_1(\tilde{u})$. The generation function of this distribution is

$$\begin{aligned} \tilde{G}_0(x) &= \frac{1}{G_0(\tilde{u})} \sum_{m=0}^{\infty} x^m \sum_{k=m}^{\infty} p_k C_k^m [G_1(\tilde{u})]^m [\tilde{u} - G_1(\tilde{u})]^{k-m} \\ &= \frac{G_0(\tilde{u} + (x-1)G_1(\tilde{u}))}{G_0(\tilde{u})}. \end{aligned} \quad (3.33)$$

The generation function for the excess degree of nodes in the residual graph is

$$\tilde{G}_1(x) = \frac{\tilde{G}_0(x)}{\tilde{G}'_0(1)} = \frac{G_1(\tilde{u} + (x-1)G_1(\tilde{u}))}{G_1(\tilde{u})}. \quad (3.34)$$

Furthermore, the coexistence threshold T_x of two disease is obtained by the equation

$$G'_1(\tilde{u}) = 1. \quad (3.35)$$

The size of epidemic of the second disease in the residual graph C can be obtained,

$$C = 1 - \tilde{G}_0(v), v = \tilde{G}_1(v). \quad (3.36)$$

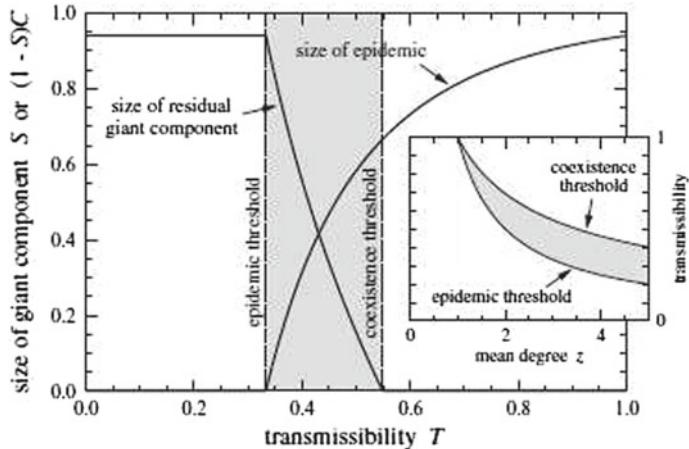


Fig. 3.3 On a graph with a Poisson degree distribution which the average degree of network is 3, the epidemic size of the first disease and the epidemic size in the residual graph versus transmissibility T [45]

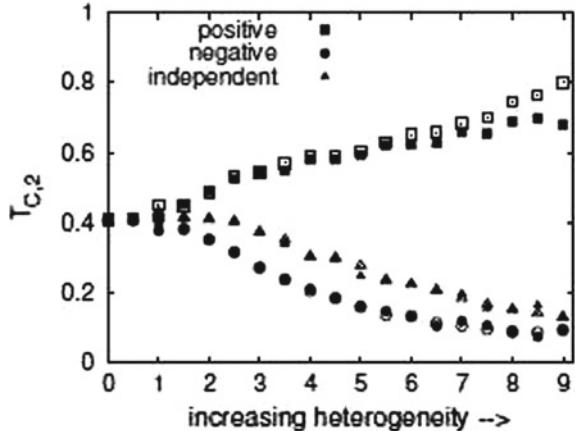
By taking the example of the network with a Poisson degree distribution, it is shown as Fig. 3.3, when T is larger than T_c , the first disease is epidemic, the size of the residual graph where the second disease is transmitted is decreasing along with T increasing. In the shaded area in the Fig. 3.3, $T_c < T < T_x$, two diseases exist simultaneously, but, at this moment, the second disease is not necessarily epidemic. Furthermore when T continues to increase and is more than $T'_c = \frac{1}{G'_1(\bar{u})}$, the second disease can be epidemic.

The interacting epidemic of two pathogen is extended to on overlay networks by Funk in 2010 [46]. In this network, there are two graphs which are denoted as Γ_1 and Γ_2 , nodes in two graphs are same, but, there are two diseases which are transmitted with transmission probabilities T_1 and T_2 in Γ_1 and Γ_2 respectively. The degree distribution of the network is given as the joint distribution $p(k_1, k_2)$, the generation function is

$$G_0^J(x, y) = \sum k_1 \sum k_2 p(k_1, k_2) x^{k_1} y^{k_2}, \quad (3.37)$$

By the similar analysis in Ref. [45], the relevant parameters of diseases are given when Γ_1 and Γ_2 don't have common edges, it implied that a first pathogen providing immunity to another disease. The impact of the transmission of the first disease on the second spreading process is illustrated by Fig. 3.4. If degrees of nodes in Γ_1 and Γ_2 are uncorrelated or negatively correlated, as heterogeneity of degrees of two network increases, the second disease is more easy to invade the residual graph of Γ_2 which is remained by removing nodes in the giant component of the first disease from Γ_2 . Conversely, if the correlation become positive, the second disease is hard to be epidemic. The above results can be generalized to the arbitrary overlap (complete

Fig. 3.4 When the average degree of the network is $\langle k \rangle = 6$, and the critical value of the first disease is $T_{C,1} = 0.25$, the critical transmission probability of the second disease $T_{C,2}$ as a function of heterogeneity when the degrees of the network are uncorrelated, negatively correlated or positive correlated [46]



overlap and partly overlapping) network. If two transmission processes affected each other, then, the possible range of coexistence or mutual exclusion depend on the joint degree distribution of the network and the extent of immunity.

Next, we summarize the application of bond percolation in a interconnected network. In 2009, Allard considered the transmission of the disease on a multitype network [47]. There are M types nodes in the network, the total number of nodes is N , and the fraction of every type nodes is $w_i (i = 1, 2, \dots, M)$. It is assumed that the degree distribution of type- i nodes is $P_i(k_1, k_2, \dots, k_M) \equiv p_i(\mathbf{k})$, then, the average number of edges from a type- i node to a type- j node is

$$z_{i,j} = \sum_{k_1=0}^{\infty} \cdots \sum_{k_M=0}^{\infty} k_j P_i(k_1, \dots, k_M) \equiv \sum_{\mathbf{k}=0}^{\infty} k_j P_i(\mathbf{k}), \quad (3.38)$$

where, $i = 1, 2, \dots, M$, $j = 1, 2, \dots, M$. The transmission probability matrix is defined as

$$\mathbf{T} = \begin{pmatrix} T_{11} & T_{12} & \cdots & T_{1M} \\ T_{21} & T_{22} & \cdots & T_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ T_{M1} & T_{M2} & \cdots & T_{MM} \end{pmatrix}, \quad (3.39)$$

where, T_{ij} is the transmission probability of the $i \rightarrow j$ edges. Using the following $2M$ generation functions of degree distribution and excess degree distribution,

$$G_i(\mathbf{x}; \mathbf{T}) = \sum_{\mathbf{k}=0}^{\infty} P_i(\mathbf{k}) \prod_{l=1}^M [1 + (x_l - 1)T_{il}]^{k_l}. \quad (3.40)$$

It is obtained that the average occupied degree is $\tilde{z}_{ij} = T_{ij}z_{ij}$, and the generation function of excess occupied degrees is

$$F_{ij}(\mathbf{x}; \mathbf{T}) = \frac{1}{\tilde{z}_{ji}} \frac{dG_j(\mathbf{x}; \mathbf{T})}{dx_i}. \quad (3.41)$$

Using formula (3.40) and (3.41), exact expression of the type outbreak size, the epidemic threshold and the epidemic size can be given. When $M = 2$, and there are edges between two different types of nodes, the network degenerate into a bipartite network, the result is different from that in Ref. [7].

The above-mentioned Refs are applications of the theory of bond percolation in the SIR model on the network. In addition, the SEIR model can be solved by the same method [48].

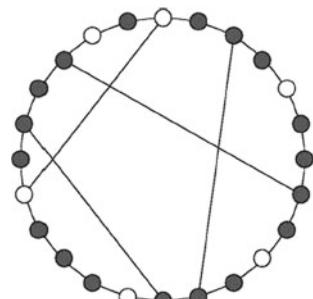
3.2.2 Site Percolation

Considering the susceptibility and transmissibility of individuals, Moore study some simple models of disease transmission on the one- dimension small-world networks [49]. In a one dimensional lattice with periodic boundary conditions as shown in Fig. 3.5, there are L sites, and the distance of every pair of sites is equal to k . Every shortcut is added with the probability ϕ , and every site is occupied with the probability p which represents the susceptibility of individuals in the process of the disease spread when L is very large, thus the probability that random two sites exist a shortcut is

$$\psi = 1 - \left(1 - \frac{2}{L^2}\right)^{k\phi L} \simeq \frac{2k\phi}{L}. \quad (3.42)$$

Local clusters in the small-world graph are some connected occupied sites by the near-neighbor bond, and a larger connected cluster consist of some local clusters connected by some shortcuts. The average number of local clusters of length i is

Fig. 3.5 This is a small-world graph with $L = 24$, $k = 1$, and four shortcuts. The gray sites which represent susceptible individuals are occupied. The susceptibility of individuals (or the occupation probability of sites) in the network is $p = \frac{3}{4}$ [49]



$$N_i = (1-p)^{2k} p [1 - (1-p)^k]^{i-1} L = (1-q)^2 p q^{i-1} L, \quad (3.43)$$

where $q = 1 - (1-p)^k$. The disease starts with some occupied site in a particular local cluster, it is transmitted and is arrived by near-neighbor bond, then other local clusters are reached by those shortcuts. The probability that a local cluster of size i has just been added to the overall connected cluster is denoted as v_i , then

$$v'_i = \sum_j M_{ij} v_j, \quad (3.44)$$

where $M_{ij} = N_i [1 - (1-\psi)^{ij}]$. The largest eigenvalue of \mathbf{M} is

$$\lambda = \psi \sum_j j_2 N_j. \quad (3.45)$$

By $\lambda = 1$, the threshold density of the epidemic is given by the formula

$$\phi = \frac{(1-p_c)^k}{2kp_c[2 - (1-p_c)^k]}, \quad (3.46)$$

p_c is a root of polynomial of order $k+1$.

If the transmission probability between an infected individual and a healthy but susceptible one is p , the disease spread can be solved by bond percolation with the occupation probability p . When $k=1$, the threshold is same with the above case of site percolation. When $k=2$, let Q_i be the probability that a given site n and its left site $n-1$ are in the same local cluster of size i , Q_{ij} denote the probability that n and $n-1$ are in two local clusters of size i and j , respectively. By defining two generating functions $H(z) = \Sigma_i Q_i z^i$ and $H(z, w) = \Sigma_{i,j} Q_{i,j} z^i w^j$, and

$$H(z, w) = z(1-p)^2 [H(w) + H(w, 1)] + zp(1-p)H(w, z), \quad (3.47)$$

and

$$H(z) = zp(2-p)H(z) + zp(1-p)H(z, 1) + zp^2H(z, z). \quad (3.48)$$

It is noted that the generating function of the cluster $G(z) = \Sigma_i N_i z^i$ satisfies

$$G(z) = (1-p)^2 [H(z) + H(z, 1)]. \quad (3.49)$$

Solving formula (3.47) and (3.48), and combining (3.49) and (3.45), it is obtained

$$\lambda = p\psi \sum_i i^2 N_i = 2k\phi p \left[\left(z \frac{d}{dz} \right)^2 G(z) \right] |_{z=1}. \quad (3.50)$$

Let $k = 2$, the percolation threshold p_c is given by the following formula

$$\phi = \frac{(1 - p_c)^3(1 - p_c + p_c^2)}{4p_c(1 + 3p_c^2 - 3p_c^3 - 2p_c^4 + 5p_c^5 - 2p_c^6)}. \quad (3.51)$$

The above method can be extended to $k > 2$.

Furthermore, for $k = 1$, sites and bonds are occupied with probabilities p_{site} and p_{bond} , the critical value can be obtained.

3.3 Epidemic Models in Complex Network with Demographics

The study of epidemic spreading in complex networks has been decades. In general, all these previous results have been obtained for closed populations, where the number of individuals is constant during the whole duration of the epidemics. The variation of population size was ignored. This is a good approximation in case such as the yearly influenza epidemics. However, for some infectious disease (e.g. HIV) with the longer spreading time, we need consider demographic processes (e.g. birth, death and migration) in the model. Demographic networks mainly affect individuals with different degrees, thus providing quantitatively and qualitatively different results.

In the following, we assume the network is uncorrelated. Each vertex represents an individual of the population and the edges represent the physical interactions among which the infection propagates. We take the susceptible-infected-susceptible (SIS) epidemic model as an example. Each individual can be either in a susceptible or infected state. Susceptible individuals become infected with probability λ if at least one of the neighbors is infected. Infected vertices, on the other hand, recover and become susceptible again with probability γ . The recovery rate for each group is the same. An individual can give birth to an individual at the same rate b . We assume that each new individual entering the network is susceptible. An individual dies at the same rate d due to natural death. An infected individual dies at the same rate α due to the infectious disease. This model allow individuals to go through the stochastic cycle susceptible→infected→susceptible by contracting the infection over and over again.

3.3.1 The Static Network with Birth and Death

If the degree distribution is given in advance, we call the static network. In the static network, if there is an empty site, an empty site may give birth to an individual. If an individual dies, there is an empty site left. In this case, demographic process is

introduced. In the network, an empty site can give birth to a healthy individual. This result can be recovered by considering the dynamical evolution of the relative density $S_k(t)$, $I_k(t)$ of susceptible and infected vertices with given degree k in the network. The SIS model in static networks with birth and death is described by the following rate equation [50, 51]

$$S_k(t) = b(1 - S_k(t) - I_k(t)) - \lambda k S_k(t) \Theta(t) - d S_k(t) + \gamma I_k(t), \quad (3.52)$$

$$I_k(t) = \lambda k S_k(t) \Theta(t) - d I_k(t) - \alpha I_k(t) - \gamma I_k(t). \quad (3.53)$$

The variable $\Theta(t)$ stand for the probability that an edge emanating from a vertex of degree k points to an infected site. For the model, the epidemic threshold depends on not only network topology but also the birth and death rate. Meanwhile, Zhang and Jin discuss the stability of the equilibrium [51]. They further analyze the model with respect to the effects of various immunization schemes.

3.3.2 The Dynamic Network with Birth and Death

As Moore and coauthors have shown, demographic dynamics alone destroys the degree distribution structure in scale-free networks, even if the newborn nodes are added as in the Barabási-Albert algorithm and the dying nodes are detached at random [52]. Hence, in real world application, the demographic process can change the network topology. The network topology becomes time dependent. This is the dynamic network. If the infectious diseases are bound to dynamic network, it would have been more realistic. Hence, understanding the effects of demography on the existence of epidemic thresholds in epidemiological models is surely important. However, the research on the epidemic model in dynamic network has been scarce [20, 53–55].

For general dynamic complex networks [4], it is assumed that new individual is distributed into group k at the probability \bar{p}_k ($0 \leq \bar{p}_k < 1$), hence $\sum_{k=1}^n \bar{p}_k = 1$. We define the attachment probability $\Pi_a(k)$ that a given edge of a newly added vertex attaches to a given preexisting vertex of degree k , and we need also consider that edges removed of other vertex of degree k due to the deaths of individuals. Let $\Pi_d(k)$ be the respective link removal probability for a node with degree k . Contributions from processes in which a vertex gains or loses two or more edges in a single unit of time will be neglected. Assume that each node has at least 1 and at most n links, $n \leq N$ (N denotes the total population). This implies that when a link is attached to a node with degree n , since any node cannot has a degree greater n , we suppose this link immediate are rewired to a preexisting vertex of degree k with $k < n$, namely, $\Pi_a(n) = 0$. Similarly, the link of a node with degree 1 is rewired when its edge is removed because of the death of another individuals, and maintain degree 1, namely, $\Pi_d(1) = 0$. In these two cases, we ignore other individual degrees changes. The evolution equations for each of these subgroups can be written as

$$\begin{aligned} S_k(t) = & B(N)\bar{p}_k - \lambda k S_k \Theta - D(N)S_k + \gamma I_k + B(N) \sum_{i=1}^n i \bar{p}_i (\Pi_a(k-1)S_{k-1} \\ & - \Pi_a(k)S_k) - D(N) \sum_{i=1}^n i N_i (\Pi_d(k)S_k - \Pi_d(k+1)S_{k+1}), \end{aligned} \quad (3.54)$$

$$\begin{aligned} I_k(t) = & \lambda k S_k \Theta - D(N)I_k - \gamma I_k + B(N) \sum_{i=1}^n i \bar{p}_i (\Pi_a(k-1)I_{k-1} \\ & - \Pi_a(k)I_k) - D(N) \sum_{i=1}^n i N_i (\Pi_d(k)I_k - \Pi_d(k+1)I_{k+1}). \end{aligned} \quad (3.55)$$

$B(N)$ denotes the recruitment rate. $D(N)$ denotes the natural death rate.

Jin et al. take $B(N) = A$, $D(N) = d$. It is assumed that a new individual links randomly those individuals already present [20]. (Similarly, it is also suitable for the preferential attachment.)

$$\Pi_a(k) = \frac{1}{N} \left(\Pi_a(k) = \frac{k}{\sum_i k_i} \right), \quad (3.56)$$

The probability that each edge of a individual is pointing to other vertex of degree k is proportional to the fraction of edges emanated from these vertices. Thus, they take

$$\Pi_d(k) = \frac{k}{\sum_i i N_i}. \quad (3.57)$$

They find that demographics has great effect on basic reproduction number. They prove that infection-free equilibrium and a unique endemic equilibrium are globally asymptotically stable. They study the impact of the demographics on the degree distribution of the population. If $A \rightarrow 0$, $d \rightarrow 0$, then the dynamic networks have become the static networks. Therefore Pastor's model is a special case of Jin's model.

Kamp has investigated susceptible-infected-death (SID) epidemic model in transmission network with natural birth and mortality [54]. Kamp's model include the death due to the disease. It is also assumed that a new individual links randomly those individuals already present. Individuals dying from natural causes at a rate d are assumed to have the same average number of contacts as found in the whole population without preferences for susceptible or infected individuals. The accurate evolution equations can be written as

$$\begin{aligned} S_k(t) = & bN\bar{p}_k - \lambda[S_kI] - dS_k + b\langle k \rangle(S_{k-1} - S_k) - d([S_kS] + [S_kI] - \\ & ([S_{k+1}S] + [S_{k+1}I])) - \alpha([S_kI] - [S_{k+1}I]), \end{aligned} \quad (3.58)$$

$$I_k(t) = \lambda[S_k I] - dI_k - \alpha I_k + b\langle k \rangle(I_{k-1} - I_k) - d([I_k S] + [I_k I] - ([I_{k+1} S] + [I_{k+1} I])) - \alpha([I_k I] - [I_{k+1} I]). \quad (3.59)$$

Involving deconvolution of individuals

$$[A_k B] \approx [AB] \frac{k A_k}{\sum_l l[A_l]}, \quad (3.60)$$

The model (3.58) and (3.59) can be simplified as follows

$$S_k(t) = bN\bar{p}_k - \lambda p_{SI}kS_k - dS_k + b\langle k \rangle(S_{k-1} - S_k) - d(kS_k - (k+1)S_{k+1}) - \alpha p_{SI}(kS_k - (k+1)S_{k+1}), \quad (3.61)$$

$$I_k(t) = \lambda p_{SI}kS_k - dI_k - \alpha I_k + b\langle k \rangle(I_{k-1} - I_k) - d(kI_k - (k+1)I_{k+1}) - \alpha p_{SI}(kI_k - (k+1)I_{k+1}). \quad (3.62)$$

where $p_{SI} = [SI] \frac{kS_k}{\sum_l l[S_l]}$, $p_{II} = [II] \frac{kI_k}{\sum_l l[I_l]}$. At last, using the method of Probability Generating Function (PGF), the model is closed. This process can be extended to the model with several stages of disease before death.

For susceptible-infected-recovered (SIR) or susceptible-infected-recovered-susceptible (SIRS) epidemic model, Volz and Meyers develop a mathematical approach to predicting disease transmission on dynamic networks in which each individual has a characteristic behavior (typical contact number), but the identities of their contacts change in time [53]. They show that dynamic contact patterns shape epidemiological dynamics in ways that cannot be adequately captured in static network models or mass-action models. Their model interpolates smoothly between static network models and mass-action models using a mixing parameter, thereby providing a bridge between disparate classes of epidemiological models. Piccardi and coauthors study SIR(S) epidemic models in network with vital dynamics of the hosts [55]. The epidemiological state of each individual may depend both on the number of contacts and on its age. They find that the result of highly connected individuals at the highest risk of infection is not as general commonly believed. They find also that SIR and SIRS processes can persist if and only if their contact rates are above a finite threshold.

3.4 Epidemic Spreading on Multilayer Networks

The last decade has witnessed a significant development in the study of epidemic processes on complex networks [56]. However, a vast majority of previous network-based epidemiological models have focused on individual networks treated as isolated systems, despite the fact that many real-world networks interact with but also depend on each other [57]. More importantly, the coupling and dependencies between

individual networks have become increasingly strong with the rapid development of technology and, therefore, modern systems should be modelled as multilayer networks [44, 58–62], or networks of networks [63–66]. In such a case, each individual network is only one part of a larger overall network—namely, is an element of a network of networks. Hence the study of single isolated networks may result in limited understanding for dynamical processes (such as percolation, diffusion dynamics and epidemic spreading) taking place on complex networks. For example, the percolation properties of a network of interdependent networks were found to be radically different from those of single isolated networks [57]. In particular, interdependent networks with scale-free degree distributions are vulnerable to random failure, which is opposite to how a single network behaves [67].

Depending on the detailed definition of different network layers and the interactions among them, the modelling of multilayer networks (or networks of networks) generally lies in the framework of interdependent [64, 67, 69], interacting [70, 71], interconnected [72], overlay [46], multiplex [73–75] networks, and many other related ones. No matter in which case, the structure and function of the overall network is much more complicated than (and not simply) the accumulation of all its layers. As long as infectious disease spreading is concerned, each network layer has its own relevance, therefore it is of great interest to generalize the classic compartmental epidemic (such as the SIS and SIR) models to these multilayer networks. In this context, two main types of multilayer networks have caught close attention in recent years: interconnected networks [68, 76, 77] and multiplex networks [73, 78–81]. In the framework of interconnected network (see Fig. 3.6), the different network layers are explicitly modelled as separate networks and the connections among them are inter-layer interactions; whereas in the framework of multiplex networks (see Fig. 3.7), different network layers are characterised in terms of multiple types of links [44, 57]. Study of epidemic processes on interconnected or multiplex networks is still in its infancy, yet is quickly growing over the years. In this section, we overview recent studies on this topic; however the literature is expected to expand rapidly thus is by no means exhaustive.

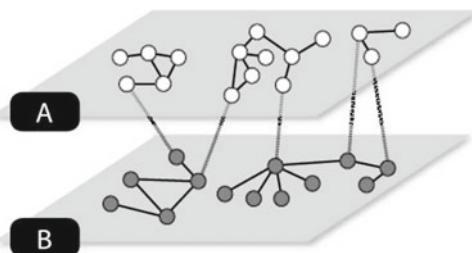
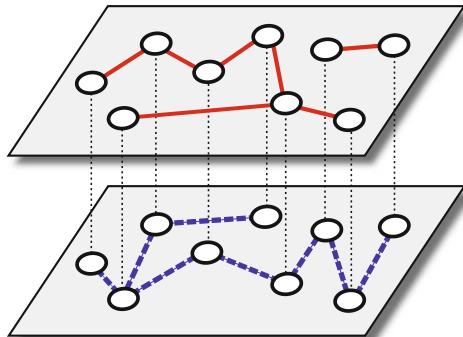


Fig. 3.6 Schematic illustration of an interconnected network system composed of two network layers: A and B. Nodes have intranetwork links within their own network layer but also internetwork links connecting them to the other network layer. Figure adapted from Ref. [68]

Fig. 3.7 (Colour online)
Schematic illustration of a multiplex network of nine nodes with two layers: the *red* (solid) and the *blue* (dashed) layer. Nodes and their replica are denoted by the *dotted line*, showing the identity relation. Figure adapted from Ref. [75]



3.4.1 Epidemic Spreading on Interconnected Networks

In this case, the spreading of an infectious disease is considered to propagate in multiple network layers, each node of each layer being a unique agent. One good example pertaining to this case is the sexually transmitted disease spreading among human population. There are three network layers of sexual contacts: one consisting of agents who have heterosexual contacts and two others composed of agents who share homosexual (gay and lesbian) relationships. These three networks are interconnected through links along which one (or both) of the linked nodes is a bisexual agent [44]. In this setting, of great importance is to determine how the epidemic dynamic behavior in the whole network relates to those in the single isolated network layers.

Several novel works have been conducted in an effort to this end. An earlier investigation of interconnected networks for simple models of infectious disease spreading concerns SIR dynamics [68] and includes two different interconnected network regimes: *strongly* and *weakly* coupled. The model considers the simplest case of only two interconnected networks of equal size, each of which follows a random Poisson degree distribution. The strongly coupled network system is defined in terms of the average internetwork connectivity, which is larger than the average intranetwork connectivity within each network layer. Otherwise, the whole network system is considered weakly coupled. In the model, each node is in one of the three possible states: susceptible (*S*), infected (*I*), or recovered (*R*). In the beginning, each node is susceptible, except for a single node in one network randomly initialized to be infected. At each time step, infected nodes infect each of their susceptible neighbours with probability β and get recovered after spending an infectious time period $t_r = 1/\mu$. The stationary states of the system are different for the two distinct interconnected network regimes. In the strongly coupled system, all network layers are simultaneously either disease free or part of an epidemic across the entire system, with the presence of interconnections enhancing epidemic spreading. In the weakly coupled network system, a new “mixed” phase appears, where the disease is epidemic in only one network, but not in the other. In other words, epidemics do not always occur across the full interconnected network system. Moreover, interconnections only affect the epidemic spreading within the less intraconnected network.

The SIS epidemic model on two interconnected complex networks has also been studied via a rigorous heterogeneous mean-field approach in Ref. [76]. The parameter conditions for the emergence of a global endemic state are given and imply that an endemic state may arise in the whole interconnected network system even if the epidemic is not able to propagate in each isolated network layer and even when the number of interconnections is small. Similarly, the critical SIS epidemic threshold in two interconnected networks is determined through the so-called N -intertwined mean-field approximation in Ref. [77], where the effect of the interconnected network structure on the epidemic threshold is discussed. In the framework of SIS epidemic model also the interplay of the adaptive rewiring and the interconnected network structure has been investigated [82]. It turns out that the system of interconnected networks is more stable (less dependent on initial conditions) with the presence of more internetwork links. In addition, a new stable state (that does not exist in a single adaptive network but only in the case of weakly coupled networks) is found, in which the disease is endemic in one network but neither becomes endemic nor dies out in the other network. This matches the work on epidemic spreading in a network with two community structures [83], where the disease becomes endemic in one community but is neither endemic nor extinct in the other.

3.4.2 Epidemic Spreading on Multiplex Networks

From the structural point of view, multiplex networks are just special cases of multilayer networks in which each node is replicated in each of the layers, therefore the layers have the same number of nodes (see Fig. 3.7). The particular setting of multiplex networks provides a best platform for investigation of the dynamical interplay between different dynamical processes occurring on the same set of nodes. In this way, multiplexity allows the probability of incorporating distinct network layers for each dynamical process in consideration. Thus, one can deal with the problem of competition between the spreading of two different diseases in a single-layer network. For instance, an epidemic model with two consecutive SIR diseases spreading within a single-layer network is developed [84]. The threshold behaviour of the first pathogen remains the same as usual; however, it is difficult for the second pathogen to spread since those agents recovered from the first disease rescue immunity also to the second one. Thus, even for scale-free networks, the epidemic threshold for both diseases to exist is nonzero.

A natural extension of competing epidemics to multiplex networks takes into account the fact that different diseases may have different transmission channels. One of the first studies addressing this issue is the one [46] that considered a multiplex network composed of two network layers and a SIR model where the disease spreads in a consecutive way. In the model, the first spreading takes place in one of the network layers leaving a number of recovered nodes. These nodes are also set to be recovered in the other layer before the second spreading taking place in the second layer. It turns out that the assortative degree correlation between nodes in both layers enhances the

immunity of the network to the second spreading (in terms of increasing the effective epidemic threshold for the coexistence of both epidemics). A generalized scenario including two SIR epidemics spreading simultaneously [85] allows to explore the impact of delay between two spreading processes and the influence of complete and partial immunity. The study of interacting epidemics has also been applied to the SIS model [86, 87].

The multiplex structure allows to integrate different dynamical processes involving the same set of nodes. In this regard, epidemic process has been recently coupled with social dynamics [78, 79, 81], attempting to understand how the social awareness about the epidemic spreading influences the latter. The analysis using a microscopic Markov chain approach reveals that there emerges a metacritical point where the diffusion of awareness is able to control the onset of epidemics [78, 79]. In the study of diffusion dynamics on multiplex networks [88], the presence of multiple pathways of diffusive motion is shown to accelerate the diffusion process. These studies collectively highlight the difference of the spreading dynamics on multiplex networks from those in a single isolated network.

The majority of the previous work on epidemic processes taking place on multilayer networks considers the spreading progress in the limit of extreme time scale separation between the network topology and the spreading dynamics (the underlying network topology is considered static). For real human contact networks, however, the static network assumption is unrealistic. Contacts are changing in time, often on a time scale comparable with that of the spreading process. Real contacts thus are dynamic, with connections appearing, disappearing and being rewired with different time scales, and are better characterised in terms of *temporal* or time-varying network [89]. In this regard, it is interesting to study epidemic spreading on adaptive or time-varying multiplex networks. For example, we may consider two spreading processes, epidemic spreading and awareness spreading, in which the former takes place on the physical contact (offline) network and the latter occurs on the communication (online) network. Both of the offline and online networks form a multiplex network with two layers. During the epidemic spreading, individuals aware of the risk of infection (after collecting the information about disease through the online network) tend to protect themselves by adaptively rewiring their infected contacts in the offline network. In this situation, the threshold behaviour of the epidemic should be revisited, and some complicated dynamic properties would be expected.

References

1. Anderson, R.M., May, R.M.: *Infectious Diseases of Humans*. Oxford University Press, Oxford (1991)
2. Pastor-Satorras, R., Vespignani, A.: Epidemic spreading in scale-free networks. *Phys. Rev. Lett.* **86**(14), 3200 (2001)
3. Pastor-Satorras, R., Vespignani, A.: Epidemic dynamics in finite size scale-free networks. *Phys. Rev. E* **65**(3), 035108 (2002)

4. Boguná, M., Pastor-Satorras, R.: Epidemic spreading in correlated complex networks. *Phys. Rev. E* **66**(4), 047104 (2002)
5. Boguná, M., Pastor-Satorras, R., Vespignani, A.: Absence of epidemic threshold in scale-free networks with degree correlations. *Phys. Rev. Lett.* **90**(2), 028701 (2003)
6. Pastor-Satorras, R., Vespignani, A.: Immunization of complex networks. *Phys. Rev. E* **65**(3), 036104 (2002)
7. Newman, M.E.J.: Spread of epidemic disease on networks. *Phys. Rev. E* **66**(1), 016128 (2002)
8. Newman, M.E.J.: The structure and function of complex networks. *SIAM Rev.* **45**(2), 167–256 (2003)
9. Wang, L., Dai, G.Z.: Global stability of virus spreading in complex heterogeneous networks. *SIAM J. Appl. Math.* **68**(5), 1495–1502 (2008)
10. Moreno, Y., Pastor-Satorras, R., Vespignani, A.: Epidemic outbreaks in complex heterogeneous networks. *Eur. Phys. J. B-Condens. Matter Complex Syst.* **26**(4), 521–529 (2002)
11. Wang, J., Liu, Z.: Mean-field level analysis of epidemics in directed networks. *J. Phys. A: Math. Theor.* **42**(35), 355001 (2009)
12. Meyers, L.A., Newman, M.E.J., Pourbohloul, B.: Predicting epidemics on directed contact networks. *J. Theor. Biol.* **240**(3), 400–418 (2006)
13. Zhang, X., Sun, G.Q., Zhu, Y.X., et al.: Epidemic dynamics on semi-directed complex networks. *Math. Biosci.* **246**(2), 242–251 (2013)
14. Wang, L., Li, X.: Spatial epidemiology of networked metapopulation: an overview. *Chin. Sci. Bull.* **59**(28), 3511–3522 (2014)
15. Boguñá, M., Castellano, C., Pastor-Satorras, R.: Nature of the epidemic threshold for the susceptible-infected-susceptible dynamics in networks. *Phys. Rev. Lett.* **111**(6), 068701 (2013)
16. Zhang, J., Jin, Z., Chen, Y.: Analysis of sexually transmitted disease spreading in heterosexual and homosexual populations. *Math. Biosci.* **242**(2), 143–152 (2013)
17. Wang, Y., Jin, Z., Yang, Z., et al.: Global analysis of an SIS model with an infective vector on complex networks. *Nonlinear Anal.: Real World Appl.* **13**(2), 543–557 (2012)
18. Kiss, I.Z., Green, D.M., Kao, R.R.: The effect of contact heterogeneity and multiple routes of transmission on final epidemic size. *Math. Biosci.* **203**(1), 124–136 (2006)
19. Lindquist, J., Ma, J., Van den Driessche, P., et al.: Effective degree network disease models. *J. Math. Biol.* **62**(2), 143–164 (2011)
20. Jin, Z., Sun, G., Zhu, H.: Epidemic models for complex networks with demographics. *Math. Biosci. Eng.* **11**(6), 1295–1317 (2014)
21. Keeling, M.J.: The effects of local spatial structure on epidemiological invasions. *Proc. R. Soc. Lond. Ser. B: Biol. Sci.* **266**(1421), 859–867 (1999)
22. Miller, J.C., Kiss, I.Z.: Epidemic spread in networks: existing methods and current challenges. *Math. Model. Nat. Phenom.* **9**(2), 4 (2014)
23. Keeling, M.J., Rand, D.A., Morris, A.J.: Correlation models for childhood epidemics. *Proc. R. Soc. Lond. Ser. B: Biol. Sci.* **264**(1385), 1149–1156 (1997)
24. Sharkey, K.J., Fernandez, C., Morgan, K.L., et al.: Pair-level approximations to the spatio-temporal dynamics of epidemics on asymmetric contact networks. *J. Math. Biol.* **53**(1), 61–85 (2006)
25. Eames, K.T.D.: Modelling disease spread through random and regular contacts in clustered populations. *Theor. Popul. Biol.* **73**(1), 104–111 (2008)
26. Taylor, M., Simon, P.L., Green, D.M., et al.: From Markovian to pairwise epidemic models and the performance of moment closure approximations. *J. Math. Biol.* **64**(6), 1021–1042 (2012)
27. Gross, T., D'Lima, C.J.D., Blasius, B.: Epidemic dynamics on an adaptive network. *Phys. Rev. Lett.* **96**(20), 208701 (2006)
28. Eames, K.T.D., Keeling, M.J.: Modeling dynamic and network heterogeneities in the spread of sexually transmitted diseases. *Proc. Natl. Acad. Sci.* **99**(20), 13330–13335 (2002)
29. Eames, K.T.D., Keeling, M.J.: Monogamous networks and the spread of sexually transmitted diseases. *Math. Biosci.* **189**(2), 115–130 (2004)
30. House, T., Keeling, M.J.: Insights from unifying modern approximations to infections on networks. *J. R. Soc. Interface rsif20100179* (2010)

31. Simon, P.L., Kiss, I.Z.: Super compact pairwise model for SIS epidemic on heterogeneous networks (2015). [arXiv:1503.01090](https://arxiv.org/abs/1503.01090)
32. Volz, E.: SIR dynamics in random networks with heterogeneous connectivity. *J. Math. Biol.* **56**(3), 293–310 (2008)
33. Miller, J.C.: A note on a paper by Erik Volz: SIR dynamics in random networks. *J. Math. Biol.* **62**(3), 349–358 (2011)
34. Miller, J.C., Slim, A.C., Volz, E.M.: Edge-based compartmental modelling for infectious disease spread. *J. R. Soc. Interface* **9**(70), rsif20110403 (2012)
35. Newman, M.E.J., Strogatz, S.H., Watts, D.J.: Random graphs with arbitrary degree distributions and their applications. *Phys. Rev. E* **64**, 026118 (2001)
36. Newman, M.E.J.: Properties of highly clustered networks. *Phys. Rev. E* **68**(2), 026121 (2003)
37. Gleeson, J.P., Melnik, S., Hackett, A.: How clustering affects the bond percolation threshold in complex networks. *Phys. Rev. E* **81**(6), 066114 (2010)
38. Miller, J.C.: Spread of infectious disease through clustered populations. *J. R. Soc. Interface*, 2009 rsif.0524 (2008)
39. Miller, J.C.: Percolation and epidemics in random clustered networks. *Phys. Rev. E* **80**(2), 020901 (2009)
40. Coupechoux, E., Lelarge, M.: How clustering affects epidemics in random networks. *Adv. Appl. Probab.* **46**(4), 985–1008 (2014)
41. Newman, M.E.J.: Random graphs with clustering. *Phys. Rev. Lett.* **103**(5), 058701 (2009)
42. Karrer, B., Newman, M.E.J.: Random graphs containing arbitrary distributions of subgraphs. *Phys. Rev. E* **82**(6), 066118 (2010)
43. Wang, B., Cao, L., Suzuki, H., et al.: Impacts of clustering on interacting epidemics. *J. Theor. Biol.* **304**, 121–130 (2012)
44. Boccaletti, S., Bianconi, G., Criado, R., et al.: The structure and dynamics of multilayer networks. *Phys. Rep.* **544**(1), 1–122 (2014)
45. Newman, M.E.J.: Threshold effects for two pathogens spreading on a network. *Phys. Rev. Lett.* **95**(10), 108701 (2005)
46. Funk, S., Jansen, V.A.A.: Interacting epidemics on overlay networks. *Phys. Rev. E* **81**(3), 036118 (2010)
47. Allard, A., Noël, P.A., Dubé, L.J., et al.: Heterogeneous bond percolation on multitype networks with an application to epidemic dynamics. *Phys. Rev. E* **79**(3), 036113 (2009)
48. Gandolfi, A.: Percolation Methods for SEIR Epidemics on Graphs. Dynamic Models of Infectious Diseases, pp. 31–58. Springer, New York (2013)
49. Moore, C., Newman, M.E.J.: Epidemics and percolation in small-world networks. *Phys. Rev. E* **61**(5), 5678 (2000)
50. Liu, J., Tang, Y., Yang, Z.R.: The spread of disease with birth and death on networks. *J. Stat. Mech.: Theory Exp.* (08), P08008 (2004)
51. Zhang, J., Jin, Z.: The analysis of an epidemic model on networks. *Appl. Math. Comput.* **217**(17), 7053–7064 (2011)
52. Moore, C., Ghoshal, G., Newman, M.E.J.: Exact solutions for models of evolving networks with addition and deletion of nodes. *Phys. Rev. E* **74**(3), 036121 (2006)
53. Volz, E., Meyers, L.A.: Susceptible-infected-recovered epidemics in dynamic contact networks. *Proc. R. Soc. B: Biol. Sci.* **274**(1628), 2925–2934 (2007)
54. Kamp, C.: Untangling the interplay between epidemic spread and transmission network dynamics. *PLoS Comput. Biol.* **6**(11), e1000984 (2010)
55. Piccardi, C., Colombo, A., Casagrandi, R.: Connectivity interplays with age in shaping contagion over networks with vital dynamics. *Phys. Rev. E* **91**(2), 022809 (2015)
56. Pastor-Satorras, R., Castellano, C., Van Mieghem, P., et al.: Epidemic processes in complex networks (2014). [arXiv:1408.2701](https://arxiv.org/abs/1408.2701)
57. D'Agostino, G., Scala, A. (eds.): Network of Networks: The Frontier of Complexity. Springer International Publishing, Switzerland (2014). doi:[10.1007/978-3-319-03518-5](https://doi.org/10.1007/978-3-319-03518-5)
58. Kurant, M., Thiran, P.: Layered complex networks. *Phys. Rev. Lett.* **96**, 138701 (2006)

59. De Domenico, M., Solé-Ribalta, A., Cozzo, E., Kivelä, M., Moreno, Y., Porter, M.A., Gómez, S., Arenas, A.: Mathematical formulation of multilayer networks. *Phys. Rev. X* **3**, 041022 (2013)
60. Kivelä, M., Arenas, A., Barthelemy, M., Gleeson, J.P., Moreno, Y.: Multilayer networks. *J. Complex Netw.* **2**, 203–271 (2014)
61. Sahneh, F.D., Scoglio, C.: Competitive epidemic spreading over arbitrary multilayer networks. *Phys. Rev. E* **89**, 062817 (2014)
62. Min, Y., Hu, J., Wang, W., Ge, Y., Chang, J., Jin, X.: Diversity of multilayer networks and its impact on collaborating epidemics. *Phys. Rev. E* **90**, 062803 (2014)
63. Gao, J., Buldyrev, S.V., Havlin, S., Stanley, H.E.: Robustness of a network of networks. *Phys. Rev. Lett.* **107**, 195701 (2011)
64. Gao, J., Buldyrev, S.V., Stanley, H.E., Havlin, S.: Networks formed from interdependent networks. *Nat. Phys.* **8**, 40–48 (2012)
65. Gao, J., Buldyrev, S.V., Stanley, H.E., Xu, X., Havlin, S.: Percolation of a general network of networks. *Phys. Rev. E* **88**, 062816 (2013)
66. Bianconi, G., Dorogovtsev, S.N.: Multiple percolation transitions in a configuration model of a network of networks. *Phys. Rev. E* **89**, 062814 (2014)
67. Buldyrev, S., Parshani, R., Paul, G., Stanley, H., Havlin, S.: Catastrophic cascade of failures in interdependent networks. *Nature* **464**(7291), 1025–1028 (2010)
68. Dickison, M., Havlin, S., Stanley, H.E.: Epidemics on interconnected networks. *Phys. Rev. E* **85**, 066109 (2012)
69. Zhou, D., Stanley, H.E., D'Agostino, G., Scala, A.: Assortativity decreases the robustness of interdependent networks. *Phys. Rev. E* **86**, 066103 (2012)
70. Leicht, E.A., D'Souza, R.M.: Percolation on interacting networks (2009). [arXiv:0907.0894](https://arxiv.org/abs/0907.0894)
71. Donges, J.F., Schultz, H.C.H., Marwan, N., Zou, Y., Kurths, J.: Investigating the topology of interacting networks. *Eur. Phys. J. B* **84**(4), 635–651 (2011)
72. Radicchi, F.: Driving interconnected networks to supercriticality. *Phys. Rev. X* **4**, 021014 (2014)
73. Lee, K.M., Goh, K.I., Kim, I.M.: Sandpiles on multiplex networks. *J. Korean Phys. Soc.* **60**(4), 641–647 (2012)
74. Battiston, F., Nicosia, V., Latora, V.: Structural measures for multiplex networks. *Phys. Rev. E* **89**, 032804 (2014)
75. Lee, K.M., Min, B., Goh, K.: Towards real-world complexity: an introduction to multiplex networks. *Eur. Phys. J. B* **88**, 48 (2015)
76. Saumell-Mendiola, A., Serrano, M., Boguñá, M.: Epidemic spreading on interconnected networks. *Phys. Rev. E* **86**, 026106 (2012)
77. Wang, H., Li, Q., D'Agostino, G., Havlin, S., Stanley, H.E., Van Mieghem, P.: Effect of the interconnected network structure on the epidemic threshold. *Phys. Rev. E* **88**, 022801 (2013)
78. Granell, C., Gómez, S., Arenas, A.: Dynamical interplay between awareness and epidemic spreading in multiplex networks. *Phys. Rev. Lett.* **111**, 128701 (2013)
79. Granell, C., Gómez, S., Arenas, A.: Competing spreading processes on multiplex networks: awareness and epidemics. *Phys. Rev. E* **90**, 012808 (2014)
80. Massaro, E., Bagnoli, F.: Epidemic spreading and risk perception in multiplex networks: a self-organized percolation method. *Phys. Rev. E* **90**, 052817 (2014)
81. Guo, Q., Jiang, X., Lei, Y., Li, M., Ma, Y., Zheng, Z.: Two-stage effects of awareness cascade on epidemic spreading in multiplex networks. *Phys. Rev. E* **91**, 012822 (2015)
82. Shai, S., Dobson, S.: Coupled adaptive complex networks. *Phys. Rev. E* **87**, 042812 (2013)
83. Peng, X.L., Small, M., Xu, X.J., Fu, X.: Temporal prediction of epidemic patterns in community networks. *New J. Phys.* **15**, 113033 (2013)
84. Newman, M.E.J.: Threshold effects for two pathogens spreading on a network. *Phys. Rev. Lett.* **95**, 108701 (2005)
85. Marceau, V., Noël, P.-A., Hébert-Dufresne, L., Allard, A., Dubé, L.J.: Modelling the dynamical interaction between epidemics on overlay networks. *Phys. Rev. E* **84**, 026105 (2011)
86. Sahneh, F.D., Scoglio, C.: May the best meme win!: new exploration of competitive epidemic spreading over arbitrary multi-layer networks (2013). [arXiv:1308.4880](https://arxiv.org/abs/1308.4880)

87. Wei, X., Valler, N., Prakash, B.A., Neamtiu, I., Faloutsos, M., Faloutsos, C.: Competing memes propagation on networks: a network science perspective. *IEEE J. Sel. Areas Commun.* **31**(6), 1049–1060 (2013)
88. Gómez, S., Díaz-Guilera, A., Gómez-Gardeñes, J., Pérez-Vicente, C.J., Moreno, Y., Arenas, A.: Diffusion dynamics on multiplex networks. *Phys. Rev. Lett.* **110**, 028701 (2013)
89. Holme, P., Saramäki, J.: Temporal networks. *Phys. Rep.* **519**, 97–125 (2012)

Chapter 4

Resilience of Spatial Networks

Daqing Li

Abstract Critical infrastructures for transmitting materials, electricity and information between distant places, can be represented as spatial networks. The resilience of spatial networks usually shows unprecedented complexity, leading to the catastrophic cascading failures in the network under various local perturbations. From the viewpoint of physics, the cascading failure process of these networks can be considered as a phase transition, which is characterized by threshold and critical exponents. In this chapter, we first review our research on the definition and measurement of the dimension of these spatial networks, which is essential for determining the critical properties of the phase transition in the network failure process according to statistical physics. Secondly, we review our research on the dynamical organization of flow on these spatial networks, which can help to locate the relation between the flow and overload in the cascading failures. Thirdly, we review our research results on the failure propagation behaviors in the cascading failures, showing long-range decay of spatial correlation between component failures. Finally, we review our research on the modeling of self-healing against cascading failures and discuss the challenges in the reliability engineering for evaluating and improving the resilience of spatial networks.

4.1 Introduction

Critical infrastructures for transmitting materials, electricity and information between distant places, can be represented as spatial networks [1]. These networks, which organize and improve our daily affairs, now possess an unprecedented dimension of complexity due to the interdependence inherited from their technological, social

D. Li (✉)

School of Reliability and Systems Engineering, Beihang University, Beijing 100191,
People's Republic of China
e-mail: daqingl@buaa.edu.cn, li.daqing.biu@gmail.com

D. Li

Science and Technology on Reliability and Environmental Engineering Laboratory,
Beijing 100191, People's Republic of China

and economic properties. As recent crisis of blackouts, congestion, cyber-attacks and storm have demonstrated, local perturbation often leads to domino-like cascades of failures in a positive feedback manner as a result of this complexity. It is impossible to improve the reliability of these complex spatial networks against different perturbations without a deep understanding of the structural properties, flow dynamics and failure behaviors of these spatial networks. The lack of these knowledge in spatial networks may lead to a systematic error in estimating the underlying risk of catastrophic consequences and inefficient reliability engineering activities.

Cascading failures in different realistic networked infrastructures have attracted much scientific attention due to their devastating effects. For example, the August 2003 cascading blackout of power grid in the northeastern US and eastern Canada caused various direct economic losses more than 10 billion dollars. There are also indirect costs of social instabilities, due to the propagation of failures to other infrastructures including communications, water supply, natural gas, transportation, health services and finance systems. The vital importance of these critical infrastructures to our society motivates devotion of significant research efforts towards understanding the resilience of spatial networks, in order to develop new methods for managing the risks and mitigating devastating consequences of cascading failures.

The aim of reliability engineering is to evaluate and improve the resilience of the spatial networks. However, the cascading failures in complex spatial networks, occurring frequently in power grid, transport networks and satellite systems, exhibit complexity including emergence, propagation, chaos and so on. The relevant properties are challenging traditional reliability theory and method ranging from the evaluation of network reliability to the improvement of network reliability. Traditional reliability methods mostly assume no (or weak) correlation between component failures, while strong correlations are often observed in realistic cascading failures in spatial network including power grid and transportation systems. These strong failure correlations will lead to the problem of combinatorial explosion when we try to analyze or calculate the network reliability based on the classical reliability methods such as fault tree analysis and disjoint minimal cut-sets. Therefore, there is a lack of valid analysis and computation tools in handling the spatial network reliability due to the strong correlation in cascading failures.

Reliability theory of spatial networks, based on the study of macroscopic statistics and microscopic mechanism of system failure, should focus on the understanding of these complexities. The corresponding key technology relies on the modeling and computation of the failure behaviors in spatial networks. The theoretical framework and relevant methods should be developed to describe the failure propagation in cascading failures, quantify the corresponding risks in the uncertainty of system collapse, and manage the network ‘health’ based on the real-time information. In the development of smart grid and intelligent transportation, systems are required to predict the health condition of its own, and provide the self-healing methods to mitigate or even recover from network failure scenarios. In the future reliability-engineering framework, we hope to understand the failure behaviors of complex system, locate the vulnerable part of networks and support the application of Smart technology.

In this chapter, focusing on the resilience of spatial networks, we will review our research progress on the resilience of spatial networks including the dimension definition of complex spatial networks, flow organization on spatial network, propagation behaviors in cascading failures on networks, and modeling of self-healing method against cascading failures. Finally, we will also discuss the challenge in the reliability engineering of complex systems.

4.2 Dimension of Spatial Network

Given a one-dimensional linear chain, it will become disconnected if any of its nodes is removed. For a two-dimensional lattice, its percolation threshold is 0.5927 [2], which means that we have to remove more than 40 % of nodes to fragment the lattice. Spatial networks with different dimension can have distinct resilience properties. In statistical physics, the dimension of a system is one of the most fundamental physical properties, which can determine the phase transition behaviors and other dynamical processes of the system [3]. Especially, the universality feature of phase transition near the critical point depends crucially on the system dimension, independent of its detailed configurations [4]. In this section, we will review our research results in [5–7] on the dimension of spatial networks.

Although the network dimension is defined in topological space [8], the definition of dimension considering spatial networks in the Euclidean space is still missing. Embedded in the real space, the realistic spatial networks including power grid, Internet and airline network are found to have unique link length distribution $P(r)$ following a power law distribution [5]. While the lattice has the dimension 2, random networks without spatial constraints have infinite dimension due to its local tree structure. The dimension of spatially embedded networks is found to be between the above two limiting cases, due to the distinctive power law distribution of link length.

We study models for spatially embedded networks (Fig. 4.1) and realistic spatial networks including power grid, Internet and airline network [5]. Dimension [9, 10] can be explored using the fact that the network mass M (number of nodes) within a hypersphere of radius r scales with r as

$$M \sim r^d. \quad (4.1)$$

In the study of this scaling relation, traditional methods of measuring dimension of fractals are box counting method and cluster growing method, which considers mostly the spatial information of the object. In another hand, methods measuring the dimension of network without spatial constraints focus only on the topological properties of network. For a given spatial network, we have developed the measurement method (Fig. 4.2), which takes into account the information of both the space and the topology [5]. We consider the number of nodes within a chemical distance l corresponding to radius r , for a given origin node. In this way, we can calculate

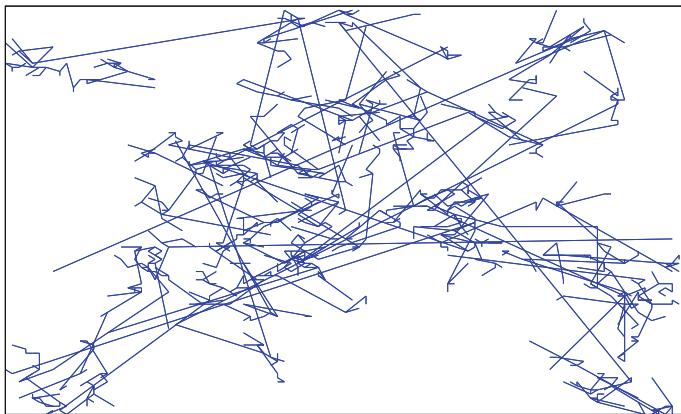
$\delta=3.5$, ER, Giant Component

Fig. 4.1 Giant component of ER networks embedded in a square lattice at the percolation threshold with $\delta = 3.5$. After [6]

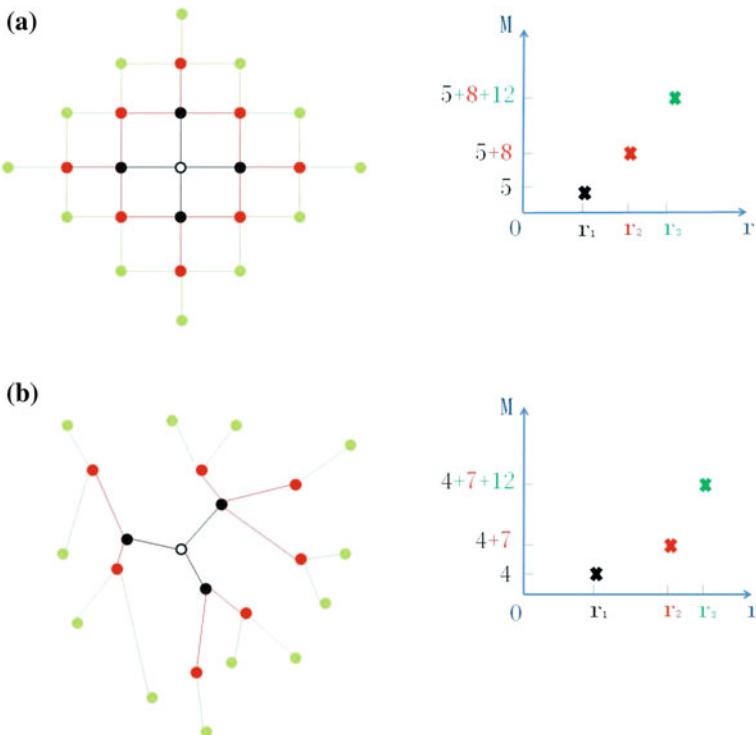


Fig. 4.2 Determination of the network dimension. **a** For a regular lattice and **b** for a complex network. The figures show, around a randomly chosen node (open circle), shell 1 (black), shell 2 (red) and shell 3 (green). On the right-hand side, the number of nodes M is plotted versus the mean spatial distance r of each shell from the origin site. From the scaling relation $M \sim r^d$, we obtain the network dimension d . After [5]

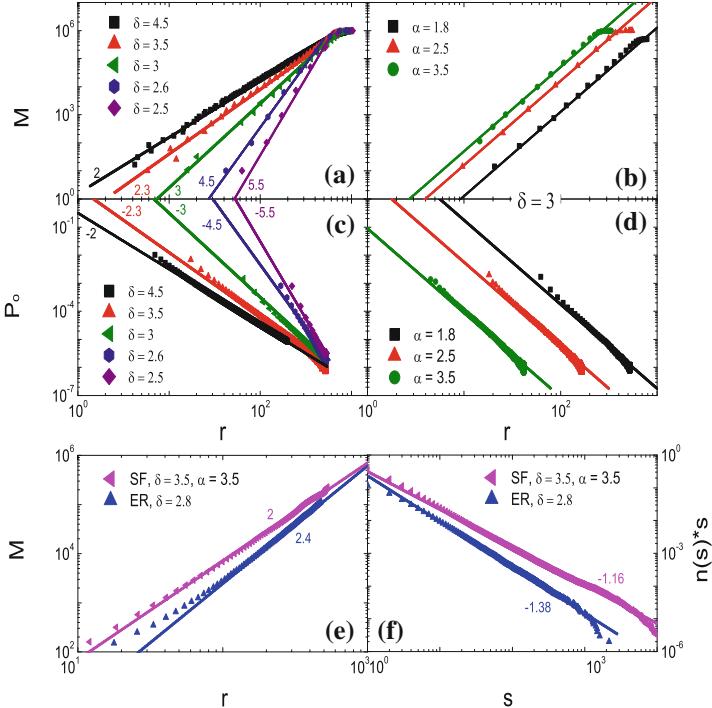


Fig. 4.3 Dimension and its relation to diffusion and percolation. The scaling relations between the mass M and the metric distance r for **a** Erdős-Rényi (ER) networks with a Poissonian degree distribution and **b** scale-free (SF) networks with a power law degree distribution ($P(k) \sim k^{-\alpha}$). In **(a)**, the distance exponent δ varies between 2.5 and 4.5. In **(b)**, $\delta = 3$ and the degree exponent α varies between 1.8 and 3.5. The figures show that the network dimension for ER networks increases monotonically with decreasing δ . For $\delta = 3$, the network dimensions in ER and scale-free networks seem to coincide. **c, d** show for the same networks as in **(a), (b)** respectively, the probability that a diffusing particle is at its starting site, after travelling an average distance r . The figure shows that the negative slopes representing the dimension in Eq. (4.1) in the double logarithmic plots agree with the network dimension obtained by Eq. (4.2). **e, f** show, for two representative scale-free and ER networks at the percolation threshold, the fractal dimension d_f (obtained from the slopes in **(e)**) and the exponent τ (obtained from the slopes of the cluster size distribution in **(f)**). The figures show that d_f and τ are related via the network dimension d : $\tau = 1 + d/d_f$. After [5]

the number of nodes $M(l)$ and the average radius $r(l)$ at a given chemical distance l . Dimension can be determined by the scaling relation between $M(l)$ and $r(l)$.

We found in [5] that spatially embedded networks can be characterized by a finite dimension (Fig. 4.3), which depends on the scaling exponent δ of link length distribution. When δ is large ($\delta > 4.5$), the dimension of spatially embedded networks is close to the dimension of the embedding space ($d_e = 2$), where the network links are mostly short ranged. Dimension increases with decreasing δ and approaches infinite when δ is becoming smaller and long range connections dominate. For realistic

spatial networks with different link length distribution, we found a dimension close to 3 for the airline network and around 4.5 for the Internet. For the power grid with an exponential distribution of link lengths, its dimension is found to around 2.

Furthermore, we investigate in [7] separately the scaling of the mass M and the Euclidean distance r with the topological distance l . Our results suggest that in the intermediate regime $d_e \leq \delta < 2d_e$, $M(l)$ and $r(l)$ do not scale with l as a power law, but as a stretched exponential: $M(l) \sim \exp[Al^{\delta'(2-\delta')}]$ and $r(l) \sim \exp[Bl^{\delta'(2-\delta')}]$, where $\delta' = \delta/d_e$. The dimension d can be calculated from parameters A and B by $d = A/B$, such that $M(l) \sim r(l)^d$. For $\delta < d_e$, M increases exponentially with l , as known in random networks for $\delta = 0$, while r is independent of l . For $\delta \geq 2d_e$, power law scaling is recovered, where $M(l) \sim l^{d_l}$ and $r(l) \sim l^{1/d_{min}}$, with $d_l \cdot d_{min} = d$.

According to statistical physics, dimension plays an important role in the processes of phase transition including percolation. The formation and disintegration processes of complex networks can be considered a typical percolation transition [11–13], where the percolation threshold and critical exponents determine the robustness features of these complex networks. With the model of spatially embedded networks having different link length distribution, we found in [6] that the percolation threshold q_c is increasing with decreasing δ . Furthermore, the critical exponents including τ and d_f are changing with different δ , suggesting that the spatial networks with different dimension belong to different universality class of percolation.

In addition to the structural properties of network, dimension is also found to influence the dynamical processes on networks [14]. Diffusion modeled by random walk is one of the widely observed phenomena including epidemics and rumor spreading. Our results in [5] suggest that for spatial networks with different dimension (Fig. 4.3), the probability P_0 that a diffusing particle returns to its origin after traveling t steps shows distinctive behavior depending on the dimension:

$$P_0 \sim r^{-d}, \quad (4.2)$$

where $r(t)$ in this case is the root mean square (r.m.s) displacement of the particle at time t . This scaling relation assumes that the probability of the particle to be in any site of the visited volume $V = r(t)^d$ is the same.

Spatial networks including Internet and airline networks are usually assumed similar as lattice. Our findings in [5–7] suggest that these spatial networks have different dimension from lattice, leading to distinct dynamical behaviors and resilience properties. This requires us to further study the function process and failure behaviors of these spatial networks and identify their intrinsic vulnerability. Especially, different percolation threshold and universality class of spatial networks suggested by their dimensions indicate the different effects in the relevant application of reliability methods.

4.3 Flow Organization on Spatial Networks

For critical infrastructures such as transportation systems, the system function means the continuous convey of freight and population from origin to destination. Due to various instability and perturbations, local flows can become overload, which may spread over and form a global cascading failures. The flow organization can determine the propagation of cascading failures to some extent. In this section, we will review our research results in [15] on the traffic flow organization.

For everyone that lives in a large city, traffic congestion is one of the most troublesome phenomena. The rapid growth of world population has further aggravated the urban traffic. For example, the population of Beijing has been growing at a breakneck speed, together with the heterogeneous spatial distribution of population, leading to a heavy burden on the city traffic. The heavy traffic congestion can have tremendous impact on various aspects ranging from travel delay, fuel consumption and pollution to even safety. For example, the consequent traffic congestion can usually cause an unexpected high cost, which is estimated to be \$186 billion for Americans in 2030. Finding solutions to traffic congestion means a vast improvement of life quality.

From the viewpoint of physics, transportation systems are considered as non-equilibrium systems composed of interacting vehicles [16–18], which undergo a phase transition between free flow and congestion. Despite of the critical role in understanding and controlling traffic congestion, very little attention has been paid to the question of how the transition is formed between congestion and free flow at the scale of city traffic network.

The traffic organization has usually been studied based on different types of models [19–26], ranging from macroscopic models based on the kinetic gas theory or fluid dynamics to microscopic approaches with equations for each car in the network. Due to the development of highways between cities and availability of real data, previous studies mostly focused on the traffic congestion on highways. For example, three-phase traffic theory developed by Kerner [25] is aiming at explaining the physics of traffic breakdown and resulting congestion on highways. According to this theory, the traffic organization on highways can be classified into three categories: free flow, synchronized flow and wide moving jam. Different from congestion on highways, isolated local flows in the traffic network can form different small clusters of high-velocity as a result of collective interactions among local flows during congestion, which will merge into global flow in the free state of traffic. Although it is essential for mitigating and controlling of city traffic organization, there is still a lack of knowledge on how the local flows in the roads are networked collectively into a global flow in a city. This question is important not only because it can deepen our understanding on the nature of city traffic, but also it may shed some light on efficient traffic control methods that have extensive benefits to the construction of future urban intelligent transportation.

There are mainly two obstacles in studying how the global flow is integrated from the local flows during the traffic congestion transition. The first obstacle comes from the lack of real data. The second obstacle is due to the lack of targeted methods to

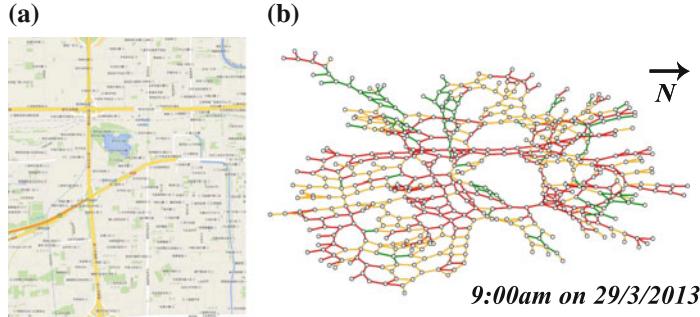


Fig. 4.4 Road network of the observed district. **a** Map of the investigated district. **b** Road network of the investigated district. Road network at 9:00 AM on March 29, 2013 is shown, where links are classified into three categories according to their velocity ratio r_{ij} : velocity ratio below 0.4 (red), between 0.4 and 0.7 (yellow), and above 0.7 (green). Note the clustering of each color. After [15]

quantify the flow organization on the network scale. To overcome the first obstacles, we collected real-time floating car data, which contains 5 min segment road velocities in the target region of Liuliqiao (Fig. 4.4) containing the largest train station in Beijing. The dataset spans almost two weeks in total. Accordingly, network of roads can be constructed, where nodes represent intersections and links represent the road segments between intersections.

We solve the second obstacles in the following way. Because roads in the region are of different levels, we employ the ratio of a given road between its current velocity and its speed limit on the same day, which is the velocity at the 95th percentile of all its velocities along the whole day, as the relative velocity of this road. In this way, dynamical function networks can be constructed from the original road network, which combined the information on both structural topology and the dynamical velocity. In this dynamical function network, nodes are still intersections and links are built based on the following equation:

$$e_{ij} = \begin{cases} 1, & r_{ij} \geq q \\ 0, & r_{ij} < q \end{cases} \quad (4.3)$$

where e_{ij} represents the existence of link in this function network. r_{ij} is the relative velocity of the link, and q is the threshold separating the link state between congested and free. This q values represent our requirement of traffic service level, and roads will be considered congested if their velocity is lower than the required service level. In this way, a function network can be constructed, which will become more diluted when q is increased.

Through tuning the value of q , the scale of traffic organization will emerge gradually: When q is 0, the function network recovers the original road network. When we increase the value of q , some links will be considered as congested if their relative velocity r_{ij} is smaller than q . Functional roads with r_{ij} higher than q form different cluster of local flows. These clusters represent high-velocity (relatively) local regions, which we can experience in our daily trip from origin to destination

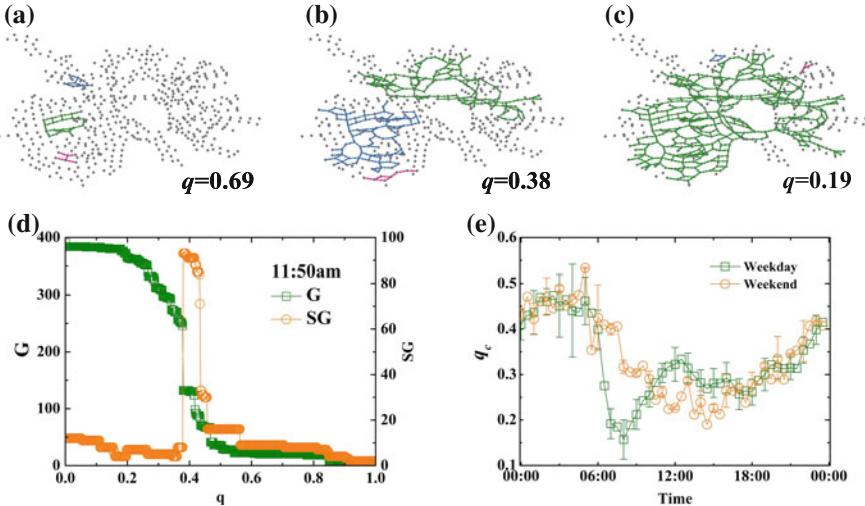


Fig. 4.5 Percolation of traffic networks: Traffic networks during the noon period (at 11:50 AM on March 27) for three q values corresponding to different connectivity states. **a, b** and **c** exhibit the traffic networks under different q values with 0.69, 0.38, and 0.19 representing the states of high-, medium-, and low-velocity thresholds, respectively. For clarity, only the largest three clusters are plotted, which are marked in green (largest cluster), blue (second-largest cluster), and strawberry (third-largest cluster). Here the clusters are strongly connected components, considering road direction. **d** Size of the largest cluster (G) and the second-largest cluster (SG) of traffic networks as a function of q . Critical value, q_c , is determined as the q value when SG becomes maximal. **e** q_c as a function of time, averaged separately over nine weekdays and two weekends. After [15]

during traffic congestion. For $q = 1$, the functional network becomes completely fragmented. Take a typical lunch time instant for example, the function networks only contain several small isolated clusters composed of roads with high relative velocity for $q = 0.69$ (Fig. 4.5a), which cannot provide good quality of global traffic service. As the value of q decreases to 0.19 (Fig. 4.5c), a giant cluster is formed with the integration of small local clusters, when we lower the requirement of traffic service level. This giant cluster extends to almost the full part of the original road network. For $q = 0.38$ (Fig. 4.5b), the size of the second largest cluster reaches its maximum, indicating the percolation process for network connectivity of functional traffic network [2, 27]. This percolation process is further investigated in Fig. 4.5d, when q is increased, the size of the second largest cluster has been increasing with the disintegration of the giant cluster, where the size of the second largest cluster reaches a maximum at the critical threshold (q_c) as the giant component is fragmented.

The value of threshold q_c measures the maximal relative velocity, above which the function network will break down. This means that a car with relative velocity above q_c will be ‘trapped’ in isolated clusters of local flows. This car needs to lower its velocity (smaller than q_c) to travel the main region of the city. Therefore, the value of threshold q_c can be considered as an effective maximal velocity for the whole region, compared with traditional limited velocity for a certain road. As Fig. 4.5e shows, q_c is at its highest value during the midnight, showing the good condition in this region.

At 6:00 am, q_c begins to drop and reaches its local minimum at 8:00 am, which reflects the increase of traffic volume during this morning rush hour. Another local minimum appearing at 18:00 is higher than the morning minimum, because the traffic is under less pressure in the evening rush hour. There is an intermediate level around 12:00 between these two local minimum. Given the different commuting habit in the weekend, only one local minimum appears around 14:00 during the weekends.

At percolation threshold q_c , the network is usually a tree-like structure and behaves as the backbone of the original network. This backbone in our defined function network plays an important role of integrating the clusters of local flow. When q is

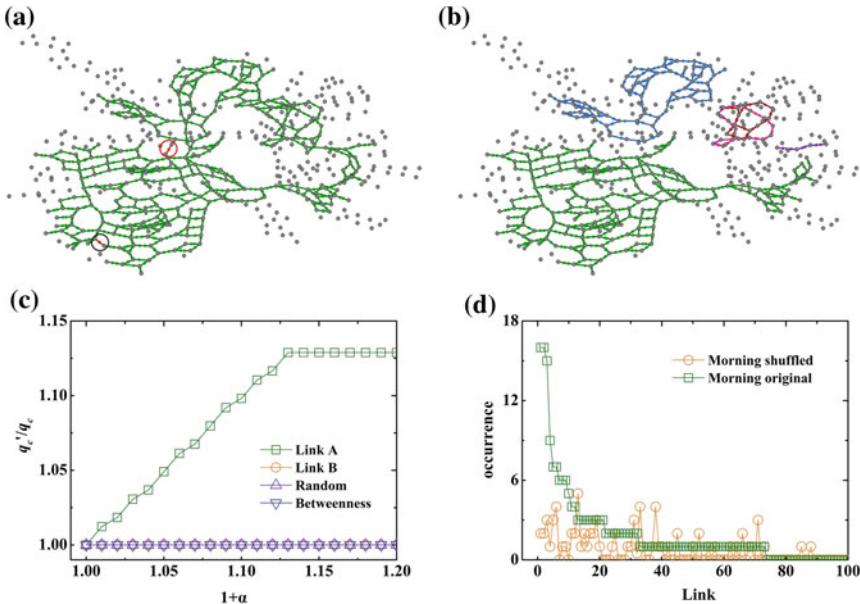


Fig. 4.6 Bottleneck links of a traffic network. **a** A typical example of a traffic network just below criticality, where two links (in red within red or black circles) are removed at criticality. Removal of them will disintegrate the giant functional network. **b** Same traffic network after removal of the two links, where the giant functional cluster is disintegrated into five clusters. We find all strongly connected clusters of the traffic network for each q and identify the links removed at threshold q_c when the second-largest strongly connected cluster reaches a maximum. Although some of these links are removed by chance, a few links do play a critical role of bridging different traffic clusters of higher velocities. These bridging links are identified as bottleneck links, because when increasing their velocity large clusters can join together to become the largest component. **c** The improvement of q_c by increasing separately the ratio ($r'_{ij} = r_{ij}(1 + \alpha)$) of two links marked in **(a)**, within which improvement of q_c can be achieved only with one (marked with red circle) of them. This link is considered a bottleneck link for global traffic. This is compared with the improvement of one link randomly chosen and the link with highest betweenness. **d** Zipf plot of occurrence times of links as bottlenecks during morning rush hours. It is compared with occurrence times of bottlenecks in the same network with shuffled values of r_{ij} during morning rush hours. For the shuffled case, we shuffle the r_{ij} values 100,000 times at each instant and find the bottleneck links with the same method. After [15]

increased slightly, this backbone will break down and the whole function network will disintegrate. Roads with different velocity can be organized as the backbone for different combinations of both network structure and traffic dynamics. The road with lowest velocity along the backbone can therefore determine the percolation threshold q_c , which is considered as the bottleneck for this functional network. As shown in Fig. 4.6a and b, the removal of the bottleneck will disintegrate the functional network into small isolated clusters of local flows. We found in Fig. 4.6c that the increase of velocity in the identified bottleneck link can improve the global organization of region traffic in terms of q_c . This improvement can hardly be achieved by the increase of velocity in other roads including the roads with highest betweenness. Compared with the bottleneck links found in the shuffled control case, different bottleneck links identified in the real traffic data reflect their own spatial features (Fig. 4.6d).

Moreover, when the identified bottlenecks are plotted in different periods of a day (Fig. 4.7a and b), evidence of different flow organization can be found. A bottleneck appearing frequently in the morning is usually replaced by other links appearing frequently in the evening. When we remove the bottleneck links found in a given period, the function network is found to disintegrate into different patterns of local flow clusters (Fig. 4.7c and d). In the morning, the function network breaks into one

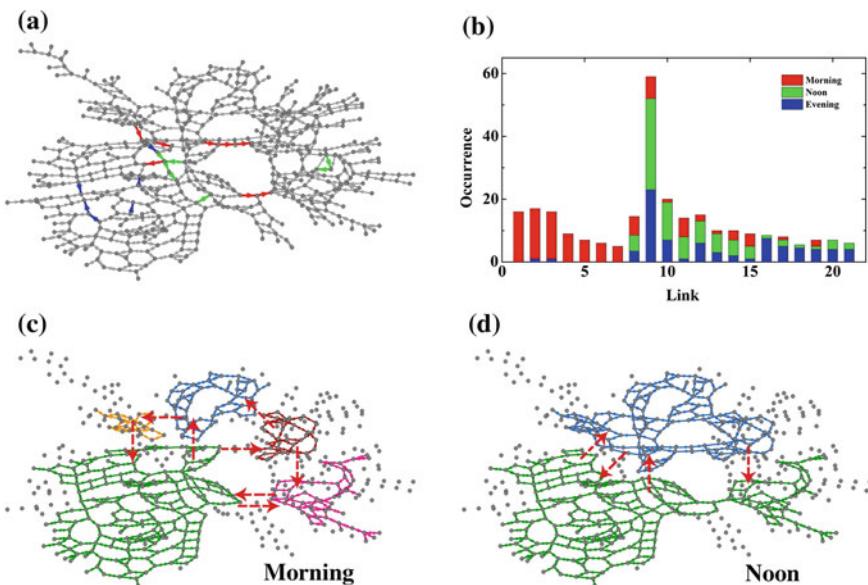


Fig. 4.7 Evolving bottlenecks in different periods in 1 d. **a** Bottleneck links with high occurrence in different periods are marked: morning (red), noon (green), and evening (blue). **b** The occurrence times of links (marked in **a**) as bottlenecks in different periods are plotted: morning (red), noon (green), and evening (blue). **c** and **d** The network breaks into several clusters after removal of bottlenecks with highest occurrence (top 10 in the morning in **c** or top 8 at noon in **d**). Red arrows in **c** and **d** are paths bridging different clusters, which are fragmented by the removal of bottleneck links. After [15]

large cluster and other four small clusters. However, at noon, the function network is disintegrated into only two clusters with similar size.

Our results in [15] show that the global traffic in the investigated region is organized in a percolation-like process, where the threshold is determined according to the velocity of the identified bottleneck. These findings can enrich the understanding of the traffic congestion on the city scale, and also may shed some light on the real-time management of city traffic. The percolation threshold of the functional network defined here can be used as an indicator showing the global organization efficiency of city traffic. The bottleneck identified in our findings may be helpful in the study of future traffic engineering.

4.4 Cascading Failures on Spatial Networks

Modern society relies on the healthy operation of various infrastructures, which are organized by large amount of components and complex interactions between them. These networked infrastructures, including transportation and power grid, have often been suffering different perturbations of random instabilities or malicious attacks. Due to the technological and social interdependence between system components, these perturbations can lead to domino-like cascades of component failures, as demonstrated in blackouts and congestions [28–31]. The global spreading of failures in the network can cause imponderable damages, possibly affecting other interdependent infrastructures [32–34]. Tremendous efforts have been made to control or mitigate the cascading failures from global propagation. However, due to its complexity and unpredictability, the frequency of large blackouts in the United States is reported not decreasing in the past decades [35].

While the existing research results are mainly focused on the critical conditions and consequences of cascading failures [36–40], the research on propagation feature of cascading failures is missing due to the lack of real data. Actually, to control efficiently the cascading failures, it is essential to study and understand the propagation behaviors of cascading failures. Particularly, due to the overloads from local perturbation, the cascading failures usually spread via hidden path instead of structural connections. This difference hinders the development of mitigation methods, which were mostly based on the network topology. In this section, we will review our relevant research results in [41].

To identify the propagation feature, in our research [41] we use the concept of spatial correlation to quantify the propagation feature of cascading failures. The spatial correlation is defined as:

$$C(r) = \frac{1}{\sigma^2} \frac{\sum_{ij, i \in F} (x_i - \bar{x})(x_j - \bar{x})\delta(r_{ij} - r)}{\sum_{ij, i \in F} \delta(r_{ij} - r)}. \quad (4.4)$$

Here x_i represents the state of component i . It is a two-value variable, which is 1 if component i is failed or 0 otherwise. \bar{x} is the average value of all x_i in the network.

σ^2 is the variance that can be calculated as $\sum_{j,j \in F} (x_j - \bar{x})^2 / N_F$, where F is the set of cascading failed components, N_F is the total number of cascading failures. $\delta(r_{ij} - r)$ is a two-value variable that selects the nodes at distance r . It is 1 if the Euclidean distance between i and j (i.e. r_{ij}) equals r , or 0 otherwise. $C(r)$ measures the spatial correlation between cascading failures, where positive values mean that the failures are tend to close to each other.

In the transportation systems, traffic congestion is formed frequently in large city, which shows the feature of cascading failures. For the realistic congestion process in the daily traffic of Beijing (see traffic information map of Beijing at morning peak shown in Fig. 4.8a, where congested roads are in red), we found that the spatial correlation between local jams exhibit the long-range behavior, which scales as a power law of geo-distance with scaling exponent around 0.6 (see Fig. 4.8c). This scaling exponent seems stable for different rush hours. Similarly, in power grid, an opened line caused by severe weather may lead to surrounding overload failures and finally result in blackouts (see the blackout occurred in the Western Systems Coordinating Council of United States shown in Fig. 4.8b, where failures are red nodes and links). In the blackouts of power grid, we also found similar long-range correlation between component faults with a larger exponent around 1.0 (see Fig. 4.8d). Despite of the differences in controlling and failure mechanisms between transportation systems and power grids, cascading failures in both spatial networks show long-range correlations, suggesting the possible universality of the propagation behavior of cascading failures.

A combined percolation-overload model is developed to investigate the possible origin of the long-range correlations in cascading failures. Using the model of cascading failures [42], we model the cascading processes of failures on spatial networks (Fig. 4.9a–d). When the model is tuned to its criticality, we found long-range correlation with exponent around 0.6 between overloads of nodes in lattices (Fig. 4.9e and f). Furthermore, when long-range connections are added in the lattice, the network topology begins to show the small-world feature of realistic power grid (Fig. 4.8b). In this way, the exponent of long-range correlation between overloads is increasing with the addition of long-range connections (Fig. 4.8d).

We further investigated the evolution of spatial correlations of cascading failures. Correlation length is defined as the length when the correlation becomes zero for the first time with increasing distance. From the realistic traffic data, we found that the correlation length increases as the number of jams increases to maximum since the free flow state of systems (Fig. 4.10a). This is also confirmed by the simulation results of the cascading failures model on the lattice (Fig. 4.10b).

Using an analogy to forest fires, if one considers each fire as a jam in traffic or a fault in power grid, a fire will spread to neighboring sites with the aid of wind and other factors. Understanding the propagation behavior of the forest fire enables to develop efficient mitigation strategies, which can guide firefighters and helicopters to isolate and eliminate the fire once it is observed. In marked contrast to forest fires, our findings reveal the long-range correlation nature between component failures in the processes of cascading failures. This finding may influence the reliability engineering efforts towards mitigating the cascading failures, which mostly assume

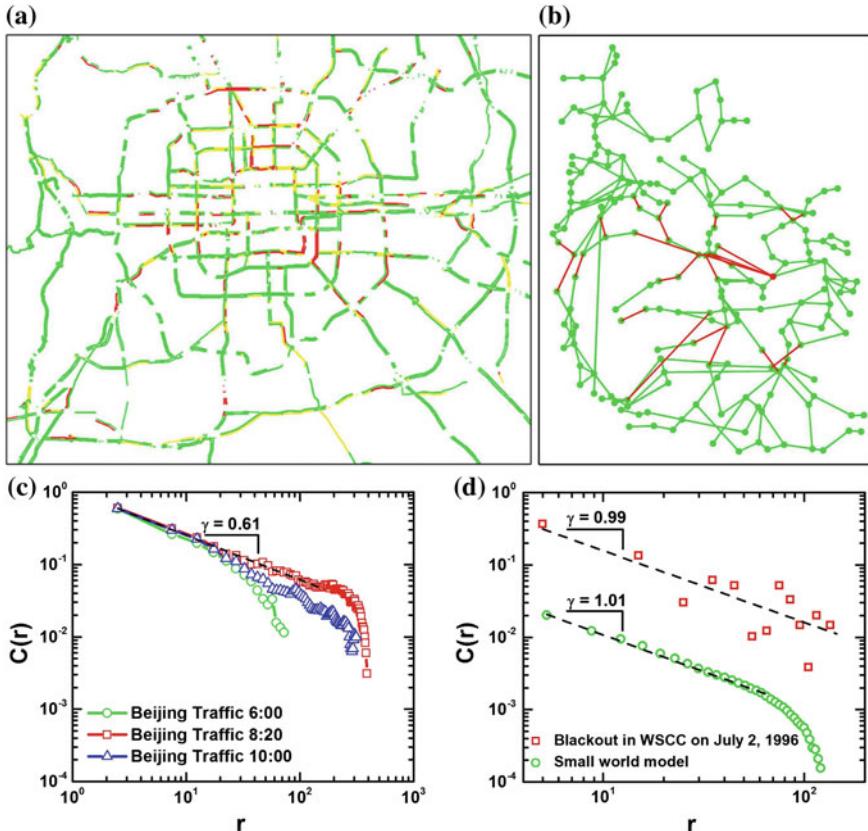


Fig. 4.8 Spatial correlations of cascading failures in real data. **a** Spatial pattern of cascading failures in traffic in Beijing, at 8:10 on 25th September 2013. Congested roads are in red (e.g. expressways with speed lower than 20 km/h), functional roads with higher speed are marked in yellow or green, depending on their velocities. **b** Spatial pattern of cascading failures in power grid (failures are red nodes and links; functional components are marked in green). This blackout occurred in the Western Systems Coordinating Council on 2nd July 1996. **c** Spatial correlation of cascading failures in city traffic for different instants. Power law spatial correlation is found at rush hour (8:20), while correlations decay faster at off-peak hours. Resolution of network distance r for congestion data is around 50 m in geographical distance. Results from 30 days in April, May, September, October and November in 2013 are averaged here. **d** Spatial correlation of cascading failures (red dots) during blackouts in power grid, where similar spatial correlation is found in the model (green dots) of cascading failures in small-world network (rewiring probability is 0.002). Resolution of network distance r for blackout data is around 5 km in geographical distance. Result in small world model (averaged over 50 realizations) is shifted down for comparison. After [41]

weak correlation (or no correlation) between failures. The corresponding isolation of local overloads in these reliability activities may fail to rescue the system from the blink of collapse, as a result of long-range correlation underlying the cascading failures. In the forest of roads and generators, network operators will need a new strategy to eliminate the risk of large-scale fires including congestions and blackouts.

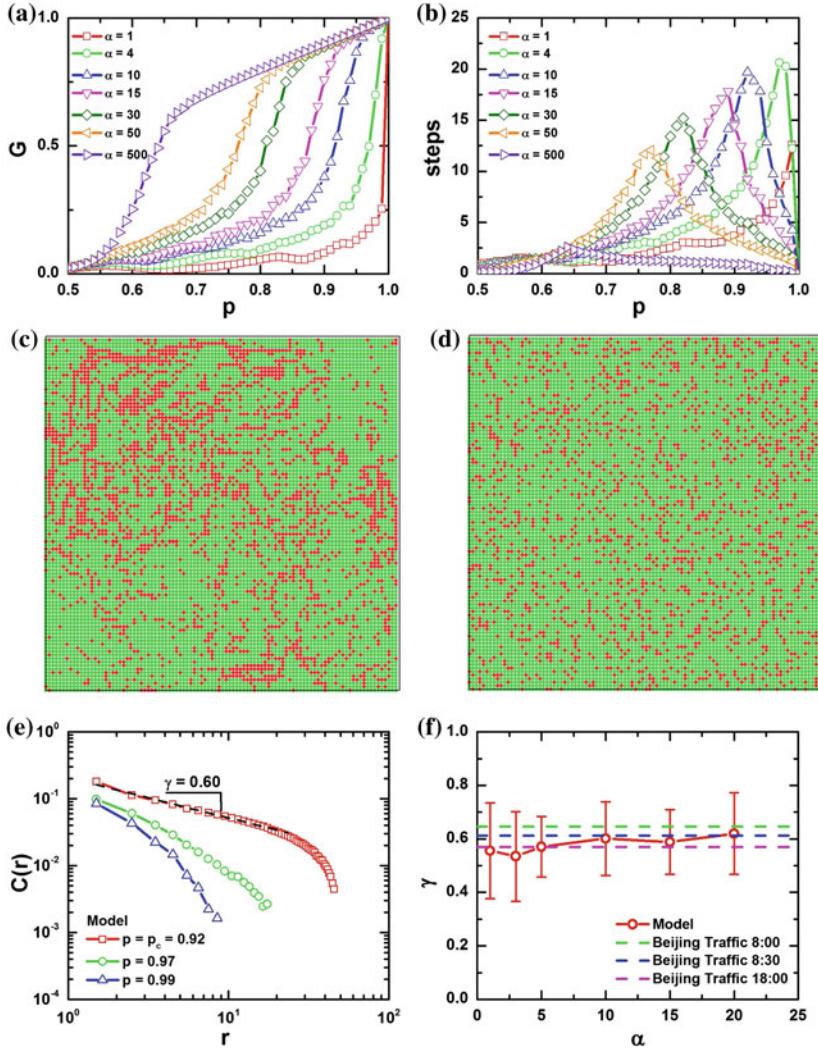


Fig. 4.9 Spatial correlations of cascading failures in model. **a** The relative size of giant component G in the network as a function of p for different α . Results are averaged over 100 realizations. **b** The number of cascading steps as a function of p for different α . The maximal cascading step corresponds to the phase transition threshold, p_c . Results are averaged over 100 realizations. **c** Spatial patterns of cascading failures in the model near criticality. One realization of model on lattice of size 10,000 is shown here, with $\alpha = 10$ and $p = 0.92$. **d** Spatial pattern of randomly distributed failures with the same number of failure as in (c). **e** Spatial correlations of cascading failures as a function of r in the model for different p values with $\alpha = 10$. Note that the correlation decays as a power law at criticality (at p_c), and decays faster when p is away from p_c . Results are averaged over 50 realizations. **f** The values of correlation exponent γ as a function of α in model and in real data of traffic during rush hours (08:00, 08:30, 18:00). Results of model are averaged over 50 realizations. Results from 30 days in April, May, September, October and November in 2013 (Beijing) are averaged in traffic data. After [41]

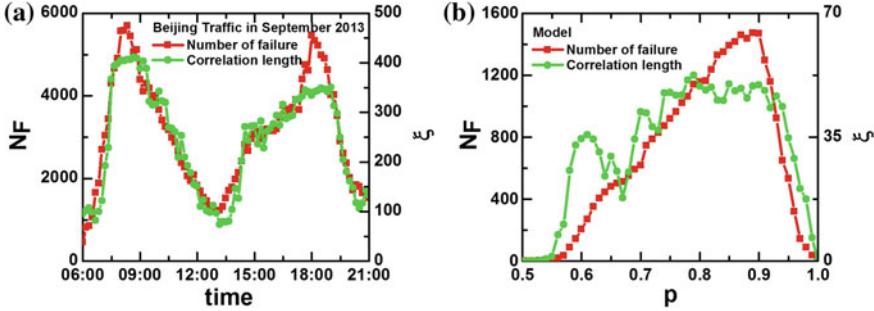


Fig. 4.10 Evolution of correlations in real traffic congestion and model. **a** Evolution of number of failures N_F and correlation length ξ in traffic data, where two maximums correspond to rush periods during a day. Results from 9 days in September (Beijing) are averaged. The maximum of ξ reaches the diameter of the main part of Beijing. **b** Evolution of number of failures N_F and correlation length ξ in the model with $\alpha = 10$, results are averaged over 10 realizations. The unit of ξ is the same as network distance r in traffic data or model. After [41]

Meanwhile, the knowledge of static correlation in the cascading failures is far from enough to understand the complexity of vulnerability and elasticity in networked infrastructures. The dynamical feature in the propagation of cascading failures should be further explored. Faced with the challenge of risk management in the future smart city, we believe that the understanding of the cascading failures will be useful for the robust control and reliable management.

Besides, in the models of network resilience the initial disturbances are often introduced as random removal or targeted attacks [43, 44]. However, spatially embedded networks are often subject to the localized attacks occurring in a geographically local region. Natural disasters and malicious attacks can be considered as typical localized attacks [45]. Considering the spatial feature of critical infrastructure networks, they are often susceptible to geographically localized damage including earthquake and tsunami. In [46], we propose a model to study the effect of localized attacks when the dependency in networks is considered. Based on the theoretical and numerical approach, we found that the damage caused by localized attacks can be larger than an equivalent random attack. Furthermore, we find that systems are metastable for a broad range of parameters. Above the critical size independent of the system size (i.e., a zero fraction), a cascading failure induced by localized attacks will spread over the whole system. These findings reveal the potential high risk of spatially embedded networks subject to localized attacks.

4.5 Restoration Against Cascading Failures

Cascading failures have become the major failure-mode in the critical infrastructures, which is not fully controlled in the corresponding reliability engineering [47–50]. In the early stage, much effort has been devoted into the reliability design, aiming

at providing the robust system architecture and enough redundancy against possible cascading failures. However, due to the unpredictable environment and various instabilities posed on system, cascading failures occurs in a higher frequency than the estimation [51].

Given the fact that cascading failures can hardly been eliminated during the design stage, research focuses have been shifted to the mitigation during the cascading failure process. Self-healing technology is one of the possible solutions with the above consideration [53–55]. The concept of self-healing technology comes from the process of recovery in animals and humans without external treatment. Self-healing technology is applied widely in many scales ranging from materials to systems.

One of the well-known examples is the application of self-healing technology in the power gird. According to North American Electric Reliability Corporation (NERC), 40 % of blackouts in America are caused by the cascading failures. Many of them is due to the mistakes made by human factors. To mitigate these cascading failures, many proactive protection systems are developed including strategic power infrastructure defense system (SPID) [56]. This system has the ability of adaptation to perturbations and fast recovery from system failures by self-healing technology. In 2003, integrated energy and communication architecture (IECSA) [57] and intelligent grid architecture are developed by EPRI, aiming at constructing a self-healing grid that can handle emergence and catastrophes. With the development of distributed generators, the ability of self-healing in the power grid will be further enhanced.

The development of self-healing technology has many aspects including hardware, software and strategy. Here we review our research results in [52, 58] on the effect of two basic parameters on the restoration effect of self-healing technology. These two parameters are the restoration timing and strength. The restoration

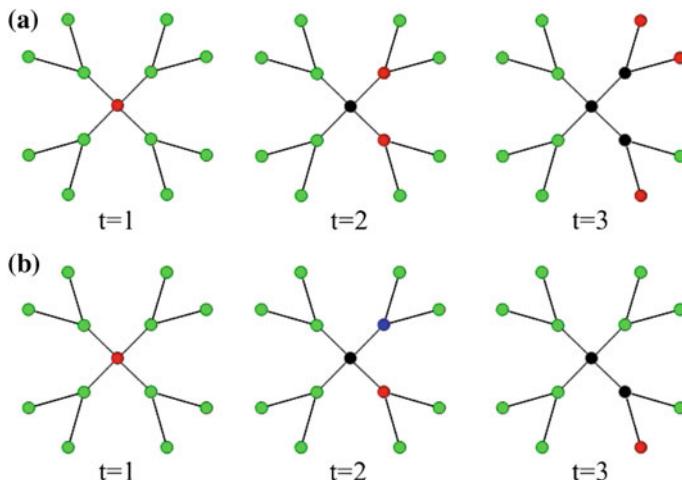


Fig. 4.11 Illustration of the self-healing model: **a** $R = 0$ and **b** $t_r = 2$, $R = 1$. The green nodes represent the functional ones, the black nodes represent the nodes failed previously, the red nodes represent the ones become failed currently and the blue nodes represent the restored ones. After [52]

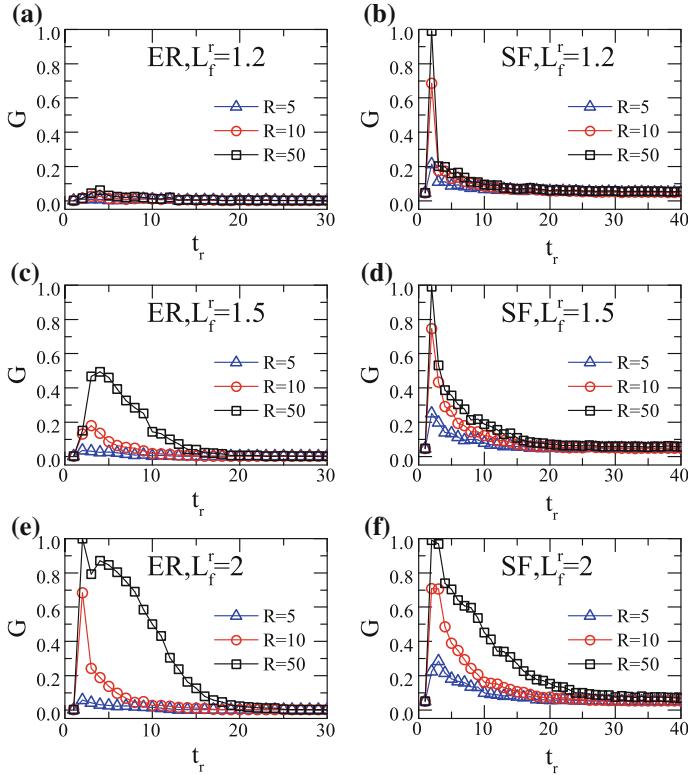


Fig. 4.12 The relative size G in ER and SF networks as a function of the restoration timing t_r : **a** ER, $L_f^r = 1.2$; **b** SF, $L_f^r = 1.2$; **c** ER, $L_f^r = 1.5$; **d** SF, $L_f^r = 1.5$; **e** ER, $L_f^r = 2$; **f** SF, $L_f^r = 2$. $D = 10$ and the initial loading of system component is $L_{min} = 0$, $L_{max} = L_f = 1$. The size of the systems is $N = 10,000$ and $\langle k \rangle = 4$. The investigated cascading processes of the network system in the manuscript have duration larger than certain values. Results are all averaged over 500 realizations. After [52]

timing t_r determines the instance when the self-healing activities are implemented. The restoration strength quantifies the maximum number of failed nodes that can be recovered during the self-healing process. An efficient self-healing method is the proper combination of these two parameters (Fig. 4.11).

Based on the cascading failure model [59], we incorporate the self-healing method to study the restoration effect. The algorithm is summarized as follows:

1. All N components are loaded uniformly in the range $[L_{min}, L_{max}]$.
2. Initially, a component i with the largest degree is disturbed with an additional load D . The component i will fail when its current load is larger than L_f , the load tolerance threshold is identical for all components.

3. Then the failed component i will transfer its current load equally to its functional neighbors. The neighbors with the transferred load may be overloaded, which will create further failures.
4. For the restoration activity during the cascades, the components failed at the chosen time step t_r will be recovered with resource R (maximum number of restored nodes) immediately after they fail. A repaired component will just recover its links to its functional neighbors. The load of a repaired component is from the load transferred to its functional neighbors when it failed. And a larger load tolerance is assigned to the restored components, that is $L_f^r > L_f$.

We use the relative size of the largest connected component in the network to quantify the restoration effect of self-healing method. As shown in Fig. 4.12, this effect depends crucially on the combination of restoration timing and strength. For ER and SF network, there can exist an optimal timing to implement the restoration at a given restoration strength. When the restoration timing is earlier than this optimal timing, the overloads are too large to be mitigated by the self-healing method. After this optimal restoration timing, the failures have already propagated over the full scale of networks, leading to the failures of a large portion of network components.

For the restoration strength, we found in Fig. 4.13 that the required number of restored nodes is increasing with the restoration timing, to keep the relative size of the largest cluster above a threshold. Similarly, when the restoration activities are postponed, the consequent state of network cannot maintain the required level, regardless of the restoration strength.

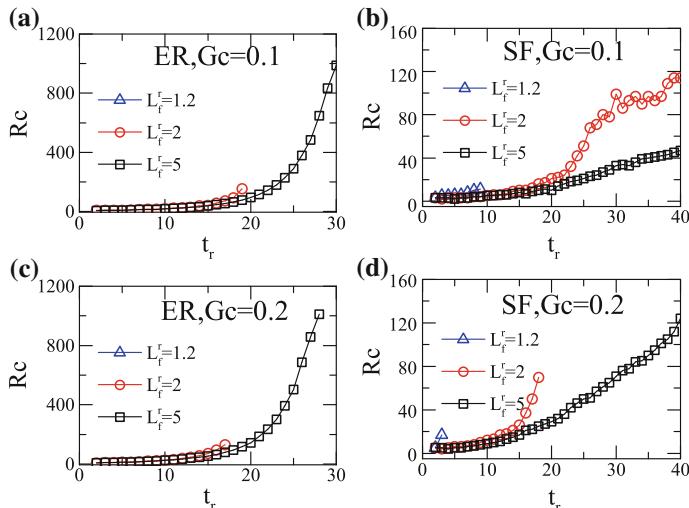


Fig. 4.13 Critical number R_c of restored resource keeping $G > G_c$ in ER and SF networks as a function of t_r : **a** ER, $G_c = 0.1$; **b** SF, $G_c = 0.1$; **c** ER, $G_c = 0.2$; **d** SF, $G_c = 0.2$. $D = 10$ and the other parameters are the same as in Fig. 4.12. Results are all averaged over 500 realizations. Notice that if no R meets the condition $G > G_c$, no point will be there. After [52]

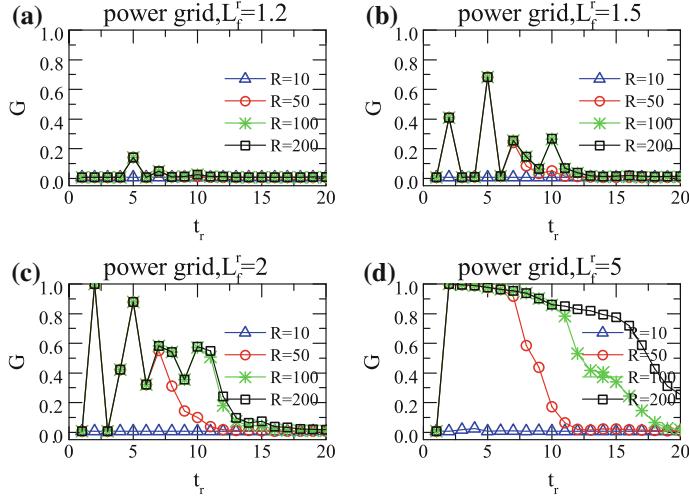


Fig. 4.14 G of the electrical power grid of the western United States [60] as a function of the restoration timing t_r : **a** $L_f^r = 1.2$; **b** $L_f^r = 1.5$; **c** $L_f^r = 2$; **d** $L_f^r = 5$. The initial loading of the system component is $L_{min} = 0.6$ and $L_{max} = L_f = 1$. The size of the systems is $N = 4941$ and the average degree $\langle k \rangle \approx 2.67$. $D = 10$ and results are all averaged over 5000 realizations. After [52]

We also tested our findings in the realistic power grid of the western United States [60]. Under targeted attacks (Fig. 4.14) or random breakdown (Fig. 4.15), the self-healing effect is found to sensitive to the restoration timing and strength, where an optimal restoration timing is also found.

Along the efforts to mitigate and control the cascading failures, the self-healing method shows its promising prospect. Although many realistic self-healing systems are still in their infancy, it is meaningful to investigate the physics behind. We found that the restoration effect is not always the best, when the self-healing method is implemented once the failures begin to cascade. There exists an optimal timing for self-healing method with a given restoration strength. Equipped with mature self-healing architectures and systems, it is critical to put them into good use based on the understanding of the effect of different restoration strategies.

4.6 Challenge in the Reliability Engineering of Complex Systems

Network reliability is the ability of network to provide required performance in a given period under given condition [61]. Our daily life assumes the availability of different reliable critical infrastructures, which can be modeled by spatial networks. Flows in different forms can traverse efficiently in a reliable network. On the other

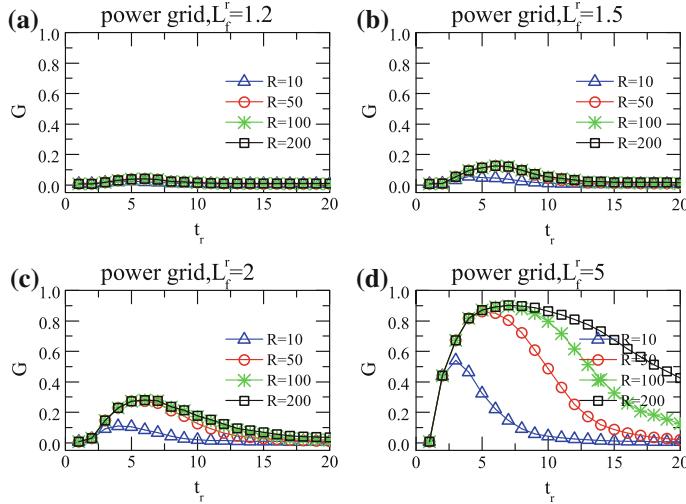


Fig. 4.15 G of the electrical power grid of the western United States [60] under random initial failure as a function of t_r : **a** $L_f^r = 1.2$; **b** $L_f^r = 1.5$; **c** $L_f^r = 2$; **d** $L_f^r = 5$. The other parameters are the same as in Fig. 4.14. Each result is averaged over 5000 realizations. After [52]

hand, the networks in practice suffer different types of internal instabilities and external perturbations, which can cause the failure of network.

In the tradition research, network reliability is modeled by terminal reliability [62–64], which considers the connectivity between the origin and destination of a given node pair. The terminal reliability can be classified into 2-end reliability, k -end reliability and full-end reliability. This method is applied in the study of reliability of networks including power grids and communication networks. However, from the viewpoint of network operators, the network reliability is not simply the sum of terminal reliability between different pairs of nodes. A reliable network in practice does not mean a complete fulfillment of full-end reliability, and also will not focus on the reliability of a specific user's terminal reliability. Especially, faced with the increasing complexity in the failure behaviors of networks, the new concept of network reliability should satisfy the following considerations:

1. The definition of network reliability should comply with the practical requirement considering both the system users and operators.
2. The failure criterion for the network should be clear. The failure criterion of networks is now mostly defined by the operators, which can be arbitrary in some occasion.
3. The computation complexity should be compatible to support the real-time evaluation of network reliability. Given a complicated network topology, the traditional methods of network reliability based on graph theory may sometimes lead to ‘combinatorial explosion’. The simulation technology of real systems is

usually required to help the calculation of network reliability, which needs the deep understanding of system details.

To achieve the above goals, we need to formulate the problem of network reliability in another way. Based on the percolation theory [50], we can shift the focus of network reliability from terminal reliability to a practical question: when the network can still be considered functional with a fraction of failed components? In the percolation theory, the failures of components are usually modeled as the removal of network nodes or links. When nodes are removed, the size of the giant component in network is decreased. At percolation threshold of removal fraction, the giant component disintegrates into small clusters, which shows typical phase transition properties. Using the methods from statistical physics, the critical point of percolation can be defined and calculated. Percolation theory has been applied in studying the robustness of single network [43, 44] and interdependent networks [32]. The failure criterion of networks can be naturally defined according to the percolation threshold of networks. Furthermore, the computation complexity of percolation properties is usually smaller than the order of N^2 .

According to the percolation theory, with the consideration of failure behaviors, we review the framework for reliability engineering of complex systems developed in [65] (Fig. 4.16).

The reliability-engineering framework can be divided into four steps:

1. Preparatory phase. In this stage, we need to collect the information and knowledge about the target system. Besides, we should identify the origin of internal instability and external perturbation. Traditional reliability analysis, including FTA (fault tree analysis) and FMECA (failure modes, effects and criticality analysis), can be performed in this stage to understand the main failure mode and cause of the failures. Special attention should be paid on the difficulties found in the traditional reliability analysis.
2. Structure and function analysis of the complex system. Based on the information and knowledge collected, we will build the network model of target systems in this stage. The network model should contain not only the structural information of system, but also the functional interdependence between components. This network model will help us to perform the reliability computation and vulnerability analysis.
3. Failure behaviors of the complex system. We begin this stage by investigating the failure mode of individual component under the perturbations. Then we will study the failure dependence between different components, using the cascading failure model. Based on the percolation theory, we will consider the operation limit of the systems and define the failure criterion of the system. The most important part of this stage is to study the failure behaviors of the targeted system, which is the basis of the next stage.
4. Reliability analysis and calculations. After we develop the reliability model based on the percolation theory, the reliability properties of the systems can be calculated with percolation method. For example, the reliability and lifetime of a system can be calculated based on the percolation threshold of the systems.

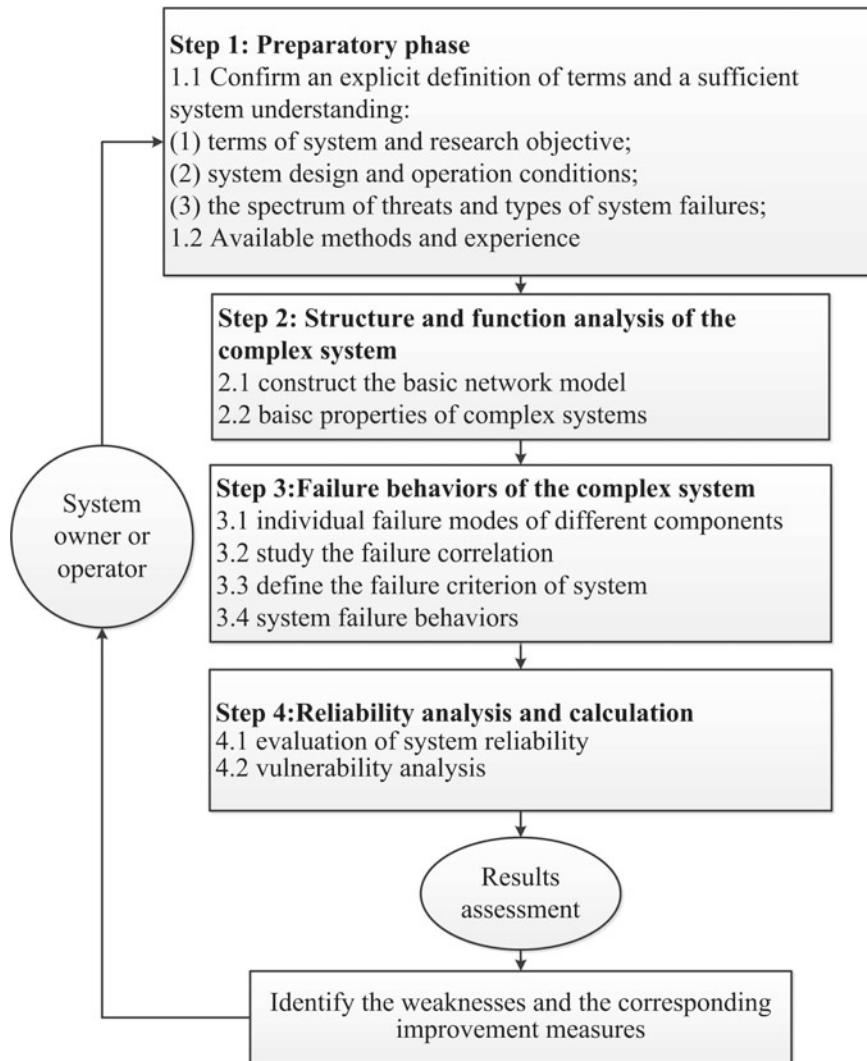


Fig. 4.16 The framework for reliability engineering of complex systems. After [65]

In a single network, the percolation threshold may be found by the following criterion [43, 44]:

$$\langle k_i | i \rightarrow j \rangle = \sum_{k_i} k_i P(k_i | i \rightarrow j) = 2, \quad (4.5)$$

where the angular brackets denote an ensemble average, k_i is the connectivity of node i , and $P(k_i | i \rightarrow j)$ is the conditional probability that node i has connectivity

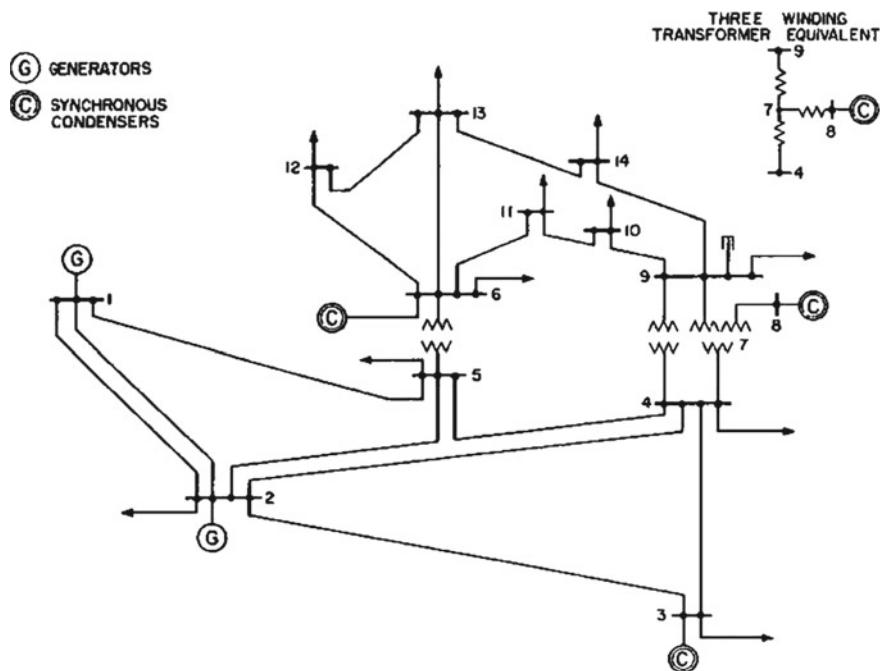
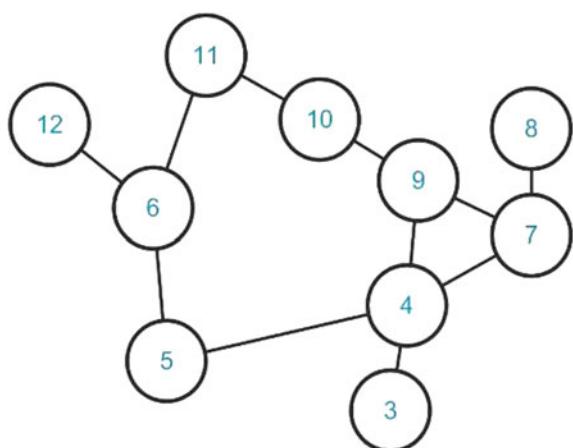


Fig. 4.17 Electrical network IEEE 14 BUS. After [66]

Fig. 4.18 Topological structure of the transmission system. After [65]



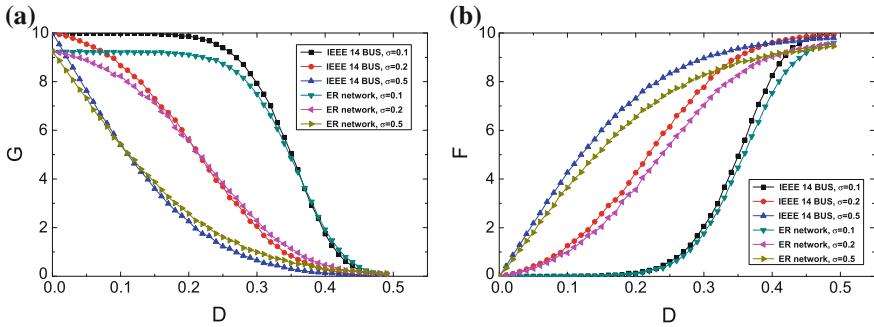


Fig. 4.19 Cascading failures as a function of disturbances in the transmission system and ER network of same size and average degree. For loads with random variables distributed with Gaussian distribution, u , the expectation of the distribution is 0.5; The parameter σ is its standard deviation. And L_{min} is 0 and L_{max} is 1. **a** Size of giant cluster, G , as a function of the disturbance, D ; **b** Total number of overloaded nodes, F , as a function of the disturbance, D . After [65]

k_i , given that it is connected to node j . The percolation threshold of interdependent network can be solved using the method in [32]. With the identified percolation threshold, together with reliability model, the reliability calculation and vulnerability analysis can be performed.

The electrical network system IEEE 14 bus (Fig. 4.17) [66] is selected as the reference case to illustrate our framework mentioned above. With the knowledge collected in the preparatory phases, we first abstract the IEEE 14 bus into a network model (Fig. 4.18). Modeling the failure behavior of IEEE 14 bus as cascading failure, we can calculate the size of giant component in network (Fig. 4.19), which can be used to quantify the system robustness.

Originating from reductionism, some traditional reliability analysis can hardly be applied into the reliability engineering of complex systems. The presented framework may help to overcome this barrier. In the meanwhile, this framework still needs further improved with the help from the network science. However, we believe that potential solutions to the challenge of complex systems lie in the combination of network science and reliability engineering.

References

1. Barthélémy, M.: Spatial networks. *Phys. Rep.* **499**(1), 1–101 (2011)
2. Bunde, A., Havlin, S.: *Fractals and Disordered Systems*. Springer, Berlin (1991)
3. Cardy, J.: *Scaling and Renormalization in Statistical Physics*. Cambridge University Press, Cambridge (1996)
4. Plischke, M., Bergersen, B.: *Equilibrium Statistical Physics*, World Scientific, Singapore (1994)
5. Li, D., Kosmidis, K., Bunde, A., Havlin, S.: Dimension of spatially embedded networks. *Nat. Phys.* **7**(6), 481–484 (2011)

6. Li, D., Li, G., Kosmidis, K., Stanley, H.E., Bunde, A., Havlin, S.: Percolation of spatially constraint networks. *EPL (Europhysics Letters)* **93**(6), 68004 (2011)
7. Emmerich, T., Bunde, A., Havlin, S., Li, G., Li, D.: Complex networks embedded in space: dimension and scaling relations between mass, topological distance, and Euclidean distance. *Phys. Rev. E* **87**(3), 032802 (2013)
8. Song, C., Havlin, S., Makse, H.: Self-similarity of complex networks. *Nature* **433**, 392–395 (2005)
9. Mandelbrot, B.B.: *The Fractal Geometry of Nature*. Macmillan (1983). ISBN 978-0-7167-1186-5. Retrieved 1 February 2012
10. Vicsek, T.: *Fractal Growth Phenomena*, pp. 31; 139–146. World Scientific, Singapore/New Jersey (1992). ISBN 978-981-02-0668-0
11. Albert, R., Barabási, A.-L.: Statistical mechanics of complex networks. *Rev. Mod. Phys.* **74**(47–97), (2002)
12. Newman, M.E.: The structure and function of complex networks. *SIAM Rev.* **45**(167–256), (2003)
13. Boccaletti, S., Latora, V., Moreno, Y., Chavez, M., Hwang, D.U.: Complex networks: structure and dynamics. *Phys. Rep.* **424**, 175–308 (2006)
14. Weiss, G.H.: *Aspects and Applications of the Random Walk*. North Holland (1994)
15. Li, D., Fu, B., Wang, Y., Lu, G., Berezin, Y., Stanley, H.E., Havlin, S.: Percolation transition in dynamical traffic network with evolving critical bottlenecks. *Proc. Natl. Acad. Sci.* **112**(3), 669–672 (2015)
16. Chowdhury, D., Santen, L., Schadschneider, A.: Statistical physics of vehicular traffic and some related systems. *Phys. Rep.* **329**(4), 199–329 (2000)
17. Helbing, D.: Traffic and related self-driven many-particle systems. *Rev. Mod. Phys.* **73**(4), 1067–1141 (2001)
18. Kerner, B.S.: *The Physics of Traffic*. Springer, Berlin (2004)
19. Lighthill, M., Whitham, G.: On kinematic waves. I. Flood movement in long rivers. *Proc. R. Soc. Lond. A* **229**(1178), 281–316 (1955)
20. Prigogine, I., Herman, R.: *Kinetic Theory of Vehicular Traffic*. Elsevier, New York (1971)
21. Newell, G.F.: A simplified theory of kinematic waves in highway traffic, part I: general theory. *Transp. Res. Part B* **27**(4), 281–287 (1993)
22. Bando, M., Hasebe, K., Nakayama, A., Shibata, A., Sugiyama, Y.: Dynamical model of traffic congestion and numerical simulation. *Phys. Rev. E Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.* **51**(2), 1035–1042 (1995)
23. Treiber, M., Hennecke, A., Helbing, D.: Congested traffic states in empirical observations and microscopic simulations. *Phys. Rev. E Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.* **62**(2 Pt A), 1805–1824 (2000)
24. Nagel, K., Schreckenberg, M.: A cellular automaton model for freeway traffic. *J. Phys. I* **2**(12), 2221–2229 (1992)
25. Kerner, B.S.: Experimental features of self-organization in traffic flow. *Phys. Rev. Lett.* **81**, 3797–3800 (1998)
26. Helbing, D., Huberman, B.A.: Coherent moving states in highway traffic. *Nature* **396**(6713), 738–740 (1998)
27. Bollobás, B.: *Random Graphs*. Academic, London (1985)
28. Toroczkai, Z., Bassler, K.E.: Network dynamics: Jamming is limited in scale-free systems. *Nature* **428**, 716 (2004)
29. Baldick, R., et al.: Initial review of methods for cascading failure analysis in electric power transmission systems IEEE PES CAMS task force on understanding, prediction, mitigation and restoration of cascading failures. In: *IEEE Power and Energy Society General Meeting-Conversion and Delivery of Electrical Energy in the 21st Century*, pp. 1–8. IEEE (2008)
30. Caldarelli, G., Chessa, A., Pammolli, F., Gabrielli, A., Puliga, M.: Reconstructing a credit network. *Nat. Phys.* **9**, 125–126 (2013)
31. Helbing, D.: Globally networked risks and how to respond. *Nature* **497**, 51–59 (2013)

32. Buldyrev, S.V., Parshani, R., Paul, G., Stanley, H.E., Havlin, S.: Catastrophic cascade of failures in interdependent networks. *Nature* **464**, 1025–1028 (2010)
33. Zhao, J.H., Zhou, H.J., Liu, Y.Y.: Inducing effect on the percolation transition in complex networks. *Nat. Commun.* **4**, 2412 (2013)
34. Radicchi, F., Arenas, A.: Abrupt transition in the structural formation of interconnected networks. *Nat. Phys.* **9**, 717–720 (2013)
35. Hines, P., Apt, J., Talukdar, S.: Trends in the history of large blackouts in the United States. In: 2008 IEEE Power and Energy Society General Meeting—Conversion and Delivery of Electrical Energy in the 21st Century, pp. 1–8. IEEE (2008)
36. Bak, P., Tang, C., Wiesenfeld, K.: Self-organized criticality—an explanation of 1/f noise. *Phys. Rev. Lett.* **59**, 381–384 (1987)
37. Watts, D.J.: A simple model of global cascades on random networks. *Proc. Natl. Acad. Sci.* **99**, 5766–5771 (2002)
38. Gao, J., Buldyrev, S.V., Stanley, H.E., Havlin, S.: Networks formed from interdependent networks. *Nat. Phys.* **8**, 40–48 (2012)
39. Lorenz, J., Battiston, S., Schweitzer, F.: Systemic risk in a unifying framework for cascading processes on networks. *Eur. Phys. J. B* **71**, 441–460 (2009)
40. Araújo, N.A., Andrade, J.S., Ziff, R.M., Herrmann, H.J.: Tricritical point in explosive percolation. *Phys. Rev. Lett.* **106**, 095703 (2011)
41. Li, D., Jiang, Y., Kang, R., Havlin, S.: Spatial correlation analysis of cascading failures: congestions and blackouts. *Sci. Rep.* **4** (2014)
42. Motter, A.E., Lai, Y.C.: Cascade-based attacks on complex networks. *Phys. Rev. Lett.* **66**, 065102 (2002)
43. Cohen, R., et al.: Resilience of the internet to random breakdowns. *Phys. Rev. Lett.* **85**, 4626–4628 (2000)
44. Cohen, R., Erez, K., Ben-Avraham, D., Havlin, S.: Breakdown of the Internet under intentional attack. *Phys. Rev. Lett.* **86**(16), 3682 (2001)
45. Neumayer, S., et al.: Assessing the impact of geographically correlated network failures. Paper Presented at Military Communications Conference, San Diego (2008). doi:[10.1109/MILCOM.2008.4753111](https://doi.org/10.1109/MILCOM.2008.4753111)
46. Berezin, Y., Bashan, A., Danziger, M.M., Li, D., Havlin, S.: Localized attacks on spatially embedded networks with dependencies. *Sci. Rep.* **5** (2015)
47. Kröger, W., Zio, E.: Vulnerable Systems. Springer, London (2011)
48. Bashan, A., Parshani, R., Havlin, S.: Percolation in networks composed of connectivity and dependency links. *Phys. Rev. E* **83**, 051127 (2011)
49. Parshani, R., Buldyrev, S.V., Havlin, S.: Interdependent networks: reducing the coupling strength leads to a change from a first to second order percolation transition. *Phys. Rev. Lett.* **105**, 048701 (2010)
50. Cohen, R., Havlin, S.: Complex Networks: Structure, Robustness and Function. Cambridge University Press, Cambridge (2010)
51. Carreras, B.A., Newman, D.E., Dobson, I., Poole, A.B.: Evidence for self-organized criticality in a time series of electric power system blackouts. *IEEE Trans. Circuits Syst. I: Regul. Pap.* **51**(9), 1733–1740 (2004)
52. Liu, C., Li, D., Fu, B., Yang, S., Wang, Y., Lu, G.: Modeling of self-healing against cascading overload failures in complex networks. *EPL (Europhysics Letters)* **107**(6), 68003 (2014)
53. Mingjun, W.: Self healing grid and distributed energy resource. *Power Syst. Technol. Beijing* **31**(6), 1 (2007)
54. Wu, D.Y., Meure, S., Solomon, D.: Self-healing polymeric materials: a review of recent developments. *Prog. Polym. Sci.* **33**(5), 479–522 (2008)
55. Sylvester, D., Blaauw, D., Karl, E.: Elastic: An adaptive self-healing architecture for unpredictable silicon. *IEEE Des. Test Comput.* **23**(6), 484–490 (2006)
56. Li, H., Rosenwald, G.W., Jung, J., Liu, C.C.: Strategic power infrastructure defense. *Proc. IEEE* **93**(5), 918–933 (2005)

57. Ming, L., Guangyu, H., Chen, S.: Brief introduction to the IECSA project. *Autom. Electr. Power Syst.* **30**(13), 99–104 (2006) (in Chinese)
58. Liu, C., Li, D., Zio, E., Kang, R.: A modeling framework for system restoration from cascading failures. *PLoS ONE* **9**(12), e112363 (2014)
59. Zio, E., Sansavini, G.: Modeling failure cascades in networks systems due to distributed random disturbances and targeted intentional attacks. In: Proceeding of the European Safety and Reliability Conference (2008)
60. Watts, D.J., Strogatz, S.H.: Collective dynamics of small-world networks. *Nature* **393**, 440–442 (1998)
61. Elsayed, E.A.: Reliability Engineering, vol. 88. Wiley (2012)
62. Wilkov, R.: Analysis and design of reliable computer networks. *IEEE Trans. Commun.* **20**(3), 660–678 (1972)
63. Wilson, J.M.: An improved minimizing algorithm for sum of disjoint products [reliability theory]. *IEEE Trans. Reliab.* **39**(1), 42–45 (1990)
64. Wood, R.K.: Factoring algorithms for computing K-terminal network reliability. *IEEE Trans. Reliab.* **35**(3), 269–278 (1986)
65. Lin, Y., Li, D., Liu, C., Kang, R.: Framework design for reliability engineering of complex systems. In: IEEE 4th Annual International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER), June 2014, pp. 19–24. IEEE (2014)
66. The IEEE 14 BUS data can be found on: <http://www.ee.washington.edu/research/pstca/>

Chapter 5

Synchronization and Control of Hyper-Networks and Colored Networks

Xinchu Fu, Zhaoyan Wu and Guanrong Chen

Abstract In this chapter, hyper-networks and colored networks corresponding to hyper-graphs and colored graphs in mathematics are presented, which can be used to model real large-scale systems, such as neuronal networks, metabolic networks, social relationship networks, scientific collaboration networks, and so on. Firstly, similarly to the BA scale-free network, both growth and preferential attachment mechanisms are adopted to generate some evolving hyper-network models, including uniform and nonuniform hyper-networks. Secondly, a uniform dynamical hyper-network model is built and its synchronization is investigated using joint-degree matrix. Thirdly, a colored network with same-dimensional node dynamics is presented. The synchronization and control of both edge-colored and colored networks are studied. Finally, a general colored network with different-dimensional node dynamics is presented and its generalized matrix projective synchronization is achieved by applying open-plus-closed-loop control.

5.1 Evolving Hyper-Network Models

Scientific collaboration networks are common, and hence familiar to researchers. In a collaboration network, the scientists can be regarded as nodes and their joint papers can be considered as edges, from which one can find the collaborative relationship between any two scientists on the network.

X. Fu (✉)

Department of Mathematics, Shanghai University, Shanghai 200444,
People's Republic of China
e-mail: xcfu@shu.edu.cn

Z. Wu

College of Mathematics and Information Science, Jiangxi Normal University,
Nanchang 330022, People's Republic of China
e-mail: wu_joy@163.com

G. Chen

Department of Electronic Engineering, City University of Hong Kong, Hong Kong,
People's Republic of China
e-mail: eegchen@cityu.edu.hk

Notice, however, that many papers have more than two coauthors. Therefore, the collaborative relationship among any three or more scientists cannot be described by the above-mentioned simple collaboration network. Another example in point is online social networks, where tripartite structure of users, resources and tags exist in the folksonomy data in online communities, which cannot be described by the above-mentioned network either.

To better describe these kinds of real networks, referred to as hyper-graphs in mathematics [1, 2], the concept of hyper-network was naturally proposed. The edges in a hyper-network, called hyper-edges, can connect to more than two nodes, which becomes useful for representing the multi-authorship collaboration networks [3, 4]. They can be used to represent the folksonomy data in online communities, in which nodes represent authors or one of the users, resources and tags, and hyper-edges represent the groups of authors that have published papers together or the three-way relationship among user, resource and tag in the folksonomy data [5]. If each hyper-edge contains exactly k nodes, the hyper-network is called a k -uniform hyper-network; otherwise, called a nonuniform hyper-network. These two types of hyper-networks will be discussed in this chapter.

In the following, the hyper-degree of node i is defined as the number of the hyper-edge attached to it, denoted by $d_h(i)$. The joint degree of nodes i_1, i_2, \dots, i_p ($p \geq 2$) is defined as the number of hyper-edges attached to these nodes, denoted by $d_{i_1 i_2 \dots i_p}$.

5.1.1 Evolving Uniform Hyper-Network Models

In the scale-free network model formulated by Barabási and Albert [6], referred to as the BA model today, two simple evolving mechanisms, i.e., growth and preferential attachment, are used to construct the network. Inspired by the scale-free network model, some generation algorithms have proposed for constructing the uniform hyper-network models, as detailed below.

Model I: In each step, there are m new nodes, one old node and one new hyper-edge

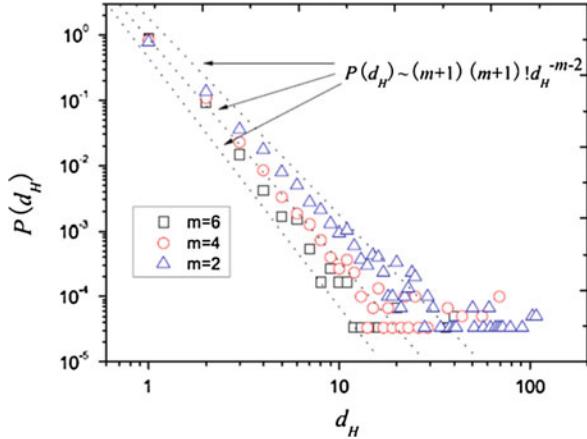
The generation algorithm of an $(m + 1)$ -uniform hyper-network is as follows [5]:

1. Initial condition: the network has a small number (m_0) of nodes and a hyper-edge which connects the initial m_0 nodes.

Then, perform the following two steps:

2. Growth: at every time step, add m nodes to the existing hyper-network, and randomly select one node from the existing hyper-network, together with the m nodes, to construct a new hyper-edge.
3. Preferential attachment: the hyper-edge E connects to a node in the existing hyper-network, where the probability $\Pi_i(d_h(i))$ for the hyper-edge E to connect to an old node i depends on the hyper-degree $d_h(i)$ of node i :

Fig. 5.1 Probability distribution $P(d_H)$ versus node hyper-edge number d_H on a logarithmic scale with different parameters. The node number in three cases with $m = 2, 4, 6$ is 10001, 10001, and 10003, respectively [5]



$$\Pi_i(d_h(i)) = \frac{d_h(i)}{\sum_{j \in N} d_h(j)}.$$

The probability distribution $P(d_h)$ of hyper-degree has the following generalized power-law form [5]:

$$P(d_h) \propto (m+1)(m+1)! d_h^{-m-2}.$$

A numerical result is shown in Fig. 5.1.

Model II: In each step, there are one new node, $k - 1$ old nodes and m new hyper-edges

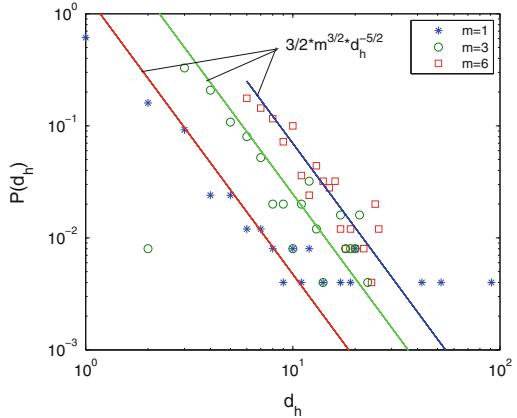
Let $d_{i_1 i_2 \dots i_{k-1}}$ denote the joint degree of nodes i_1, i_2, \dots, i_{k-1} , which is defined as the number of hyper-edges containing nodes i_1, i_2, \dots, i_{k-1} .

The generation algorithm for a k -uniform hyper-network is as follows [7]:

- Initial condition: the network has a small number m_0 ($m_0 \geq k$) of nodes and a hyper-edge which connects the initial m_0 nodes.
Then, perform the following two steps:
- Growth: add one node and m hyper-edges to the existing hyper-network, where $m \leq C_{m_0}^{k-1} = \frac{m_0!}{(k-1)!(m_0-k+1)!}$.
- Preferential attachment: the probability $\Pi_{i_1 i_2 \dots i_{k-1}}$, that a new hyper-edge contains the new node and $k - 1$ nodes selected from the existing hyper-network, depends on the joint degree of the $k - 1$ nodes at time t , in the form of

$$\Pi_{i_1 i_2 \dots i_{k-1}} = \frac{d_{i_1 i_2 \dots i_{k-1}}}{\sum_{1 \leq l_1 < l_2 < \dots < l_{k-1} \leq m_0+t-1} d_{l_1 l_2 \dots l_{k-1}}},$$

Fig. 5.2 Hyper-degree distribution $p(d_h)$ versus node hyper-degree number d_h on a logarithmic scale with different parameters. The node number is $N = 250$ [7]



where $1 \leq i_1 < i_2 < \dots < i_{k-1} \leq m_0 + t - 1$.

The probability distribution $P(d_h)$ of hyper-degree has the following generalized power-law form [7]:

$$P(d_h) \propto \frac{k}{k-1} m^{\frac{k}{k-1}} d_h^{-2 - \frac{1}{k-1}},$$

which is the same as that of the BA scale-free network with $k = 2$.

Figure 5.2 shows the hyper-degree distribution of the 3-uniform hyper-network model, with $m_0 = 4$, $m = 1, 3, 6$ respectively, and $N = 250$.

Model III: In each step, there are m_1 new node, m_2 old nodes and m new hyper-edges

The generation algorithm of a uniform hyper-network is as follows [8]:

1. Initial condition: the network has a small number (m_0) of nodes and a hyper-edge which connects the initial m_0 nodes.

Then, perform the following two steps:

2. Growth: the arrival process of nodes is a Poisson process, when m_1 new nodes enter the hyper-network at time t , m_2 old nodes are chosen to generate a new hyper-edge with the m_1 new nodes, so totally m hyper-edges are generated, where $1 \leq m, m_2 \leq m_0$.
3. Preferential attachment: the probability $\Pi_i(d_h(i))$ that a new hyper-edge E connects to an old node i depends on the hyper-degree $d_h(i)$ of node i , in the following form:

$$\Pi_i(d_h(i)) = \frac{d_h(i)}{\sum_{j \in N} d_h(j)}.$$

The probability distribution $P(d_h)$ of hyper-degree has the following generalized power-law form [8]:

$$P(d_h) \propto \frac{m_1 + m_2}{m_2} m^{\frac{m_1+m_2}{m_2}} d_h^{-2 - \frac{m_1}{m_2}}.$$

5.1.2 Evolving Nonuniform Hyper-Network Models

The generation algorithm of a nonuniform hyper-network is as follows [9]:

1. Initial condition: the network has a small number m_0 ($m_0 \geq k$) of nodes and a hyper-edge which connects the initial m_0 nodes.
Then, perform the following two steps:
2. Growth: the arrival process of nodes is a Poisson process $N(t)$ with constant λ at time t , when the $N(t)$ batch of new nodes enter the hyper-network, where the number of new nodes $\eta_{N(t)}$ obey the distribution $F(\eta)$, and m_2 old nodes are chosen to generate a new hyper-edge with the $\eta_{N(t)}$ new nodes, thus totally m hyper-edges are generated, where $1 \leq m, m_2 \leq m_0$.
3. Preferential attachment: at time t , the probability that a new hyper-edge E connects to the j th node in the i th batch of added nodes is proportional to the α th power of its hyper-degree, namely,

$$\Pi(h_j(t, t_i)) = \frac{(h_j(t, t_i))^{\alpha}}{\sum_{i,j} (h_j(t, t_i))^{\alpha}},$$

where t_i denotes the entering time of the i th batch of nodes, $0 < \alpha < 1$, $m_1 = \int \eta dF(\eta)$.

The probability distribution $P(d_h)$ of hyper-degree has the following generalized power-law form [9]:

$$P(d_h) \propto \frac{m_1}{m_2} \mu \exp\left(-\frac{m_1 \mu}{m_2(1-\alpha)} (d_h^{1-\alpha} - m^{1-\alpha})\right) d_h^{-\alpha},$$

where $\mu = \lim_{t \rightarrow \infty} \frac{1}{mm_1\lambda t} \sum_{ij} (h_j(t, t_i))^{\alpha}$.

Besides the above evolving hyper-network models, some other models have also been studied [10, 11]. In [10], a local-world evolving hyper-network model is discussed.

5.2 Synchronization on Uniform Dynamical Hyper-Networks

Real networks usually contain large numbers of interactive individuals, and the individuals have dynamical behavior described by dynamical systems. As a typical collective dynamical behavior of network with dynamical nodes, synchronization has been found important and useful in various fields, such as clock synchrony of sensor networks, rhythm of brain cells, sync time in parallel computing [12], and so on.

5.2.1 Synchronization on Linearly-Coupled Hyper-Networks

Consider a k -uniform hyper-network coupled with N identical node systems, described by [7]

$$\dot{x}_{i_1}(t) = f(x_{i_1}(t)) + \varepsilon \sum_{i_2, \dots, i_k=1}^N a_{i_1 i_2 \dots i_k} \sum_{j=2}^k \Gamma(x_{i_j}(t) - x_{i_1}(t)), \quad i_1 = 1, 2, \dots, N, \quad (5.1)$$

where i_l ($l = 1, 2, \dots, k$) are mutually different and $a_{i_1 i_2 \dots i_k}$ is defined as: if there exists a hyper-edge contains nodes i_1, i_2, \dots, i_k , then $a_{i_1 i_2 \dots i_k} = 1$; otherwise, $a_{i_1 i_2 \dots i_k} = 0$.

As an example, for $k = 3$, the hyper-network (5.1) is simplified to

$$\dot{x}_i(t) = f(x_i(t)) + \varepsilon \sum_{j=1}^N \sum_{k=1}^N a_{ijk} \Gamma(x_j(t) + x_k(t) - 2x_i(t)). \quad (5.2)$$

For simplicity, consider the synchronization of the 3-uniform hyper-network (5.2). With coupling, the network is

$$\dot{x}_i(t) = f(x_i(t)) + \varepsilon \sum_{j=1}^N \left(\sum_{k=1}^N a_{ijk} \right) \Gamma(x_j(t) - x_i(t)) + \varepsilon \sum_{k=1}^N \left(\sum_{j=1}^N a_{ijk} \right) \Gamma(x_k(t) - x_i(t)). \quad (5.3)$$

Let d_{ij} be the joint degree of nodes i and j ($j \neq i$), which is defined as the number of hyper-edges containing nodes i and j :

$$d_{ij} = \sum_{k=1}^N a_{ijk} = \sum_{k=1}^N a_{ikj} = \dots = \sum_{k=1}^N a_{kji}.$$

Rewrite (5.3) as

$$\dot{x}_i(t) = f(x_i(t)) + \varepsilon \sum_{j=1, j \neq i}^N d_{ij} \Gamma(x_j(t) - x_i(t)) + \varepsilon \sum_{k=1, k \neq i}^N d_{ik} \Gamma(x_k(t) - x_i(t)). \quad (5.4)$$

Thus, one has

$$\dot{x}_i(t) = f(x_i(t)) + 2\varepsilon \sum_{j=1, j \neq i}^N d_{ij} \Gamma(x_j(t) - x_i(t)). \quad (5.5)$$

Define the diagonal elements of D as

$$d_{ii} = - \sum_{j=1, j \neq i}^N d_{ij} = - \sum_{j=1, j \neq i}^N d_{ji}.$$

Then, Eq. (5.5) gives

$$\dot{x}_i(t) = f(x_i(t)) + 2\varepsilon \sum_{j=1}^N d_{ij} \Gamma x_j(t). \quad (5.6)$$

The hyper-network considered here is always assumed to be connected, i.e., any pair of nodes is reachable along some hyper-edges. Thus, it can be verified that the joint degree matrix $D = (d_{ij})$ is irreducible and its eigenvalues are $0 = \lambda_1 > \lambda_2 \geq \dots \geq \lambda_N$.

The objective here is for the network (5.6) to synchronize with a given orbit $s(t)$, i.e.,

$$||x_i(t) - s(t)|| \rightarrow 0 \quad (t \rightarrow \infty) \quad \forall i = 1, 2, \dots, N. \quad (5.7)$$

where $s(t)$ is a solution of an isolated node, satisfying $\dot{s}(t) = f(s(t))$. Here, $s(t)$ can be an equilibrium point, a periodic orbit, or even a chaotic attractor.

Let $x_i(t) = s(t) + \eta_i(t)$ and linearize (5.6) about $s(t)$. This leads to

$$\dot{\eta}_i(t) = Df(s(t))\eta_i(t) + 2\varepsilon \sum_{j=1}^N d_{ij} \Gamma \eta_j(t), \quad (5.8)$$

where $Df(s(t))$ is the Jacobian of f on $s(t)$. Then, referring to [13], one has the following theorems.

Theorem 5.1 ([7]) Let $0 = \lambda_1 > \lambda_2 \geq \dots \geq \lambda_N$ be the eigenvalues of the joint degree matrix $D = (d_{ij})$. If the following $N - 1$ of n -dimensional linear time-varying systems are exponentially stable:

$$\dot{\xi}(t) = (Df(s(t)) + 2\varepsilon\lambda_k \Gamma)\xi(t), \quad k = 2, 3, \dots, N, \quad (5.9)$$

then the synchronized states (5.7) are exponentially stable.

Theorem 5.2 ([7]) Suppose that there exists an $n \times n$ diagonal matrix $P > 0$ and two constants $\bar{\delta} < 0$ and $\tau > 0$, such that

$$(Df(s(t)) + \bar{\delta}\Gamma)^T P + P(Df(s(t)) + \bar{\delta}\Gamma) \leq -\tau I_n, \quad (5.10)$$

for all $\delta \leq \bar{\delta}$, where I_n is an $n \times n$ identity matrix. If

$$2\varepsilon\lambda_2 \leq \bar{\delta}, \quad (5.11)$$

then the synchronized states (5.7) are exponentially stable.

It is clear that the inequality (5.11) is equivalent to

$$\varepsilon \geq \frac{\bar{\delta}}{2\lambda_2}. \quad (5.12)$$

Therefore, the synchronizability of the 3-uniform hyper-network (5.2) with respect to a given coupling matrix can be characterized by the second-largest eigenvalue λ_2 of the joint degree matrix.

Remark 5.1 Similar to the above discussion, synchronization on a general k -uniform hyper-network can be investigated as well. In fact, referring to the definition of joint degree, the hyper-network (5.1) can be simplified to

$$\dot{x}_i(t) = f(x_i(t)) + (k-1)\varepsilon \sum_{j=1}^N d_{ij} \Gamma x_j(t).$$

Example 5.1 Consider a hyper-network consisting of 100 coupled Chua's oscillators [14] described by

$$\begin{pmatrix} \dot{s}_1 \\ \dot{s}_2 \\ \dot{s}_3 \end{pmatrix} = \begin{pmatrix} \alpha(s_2 - s_1 - f(s_1)) \\ s_1 - s_2 + s_3 \\ -\beta s_2 - \gamma s_3, \end{pmatrix}$$

where $f(s_1) = bs_1 + 0.5(a-b)(|s_1+1|-|s_1-1|)$ is a piecewise linear function, $\alpha > 0$, $\beta > 0$, $\gamma > 0$, $a < 0$ and $b < 0$.

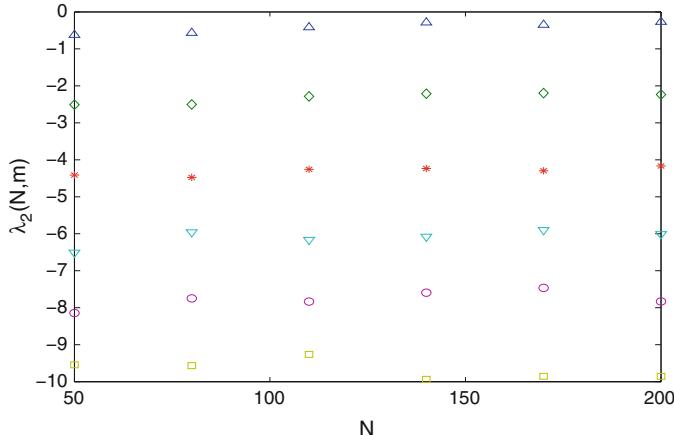


Fig. 5.3 The second-largest eigenvalue of the joint degree matrix. Here, ‘ Δ ’ are for $m = 1$, ‘ \diamond ’ for $m = 2$, ‘*’ for $m = 3$, ‘ ∇ ’ for $m = 4$, ‘ \circ ’ are for $m = 5$ and $m = 6$

Figure 5.3 shows the second-largest eigenvalues of the joint degree matrix with different pairs of N and m , where $\lambda_2(N, m)$ is obtained by averaging the results of 10 runs. In this example, $\lambda_2(N, 3) < -4$. In the numerical simulation, choose $\alpha = 10$, $\beta = 15$, $\gamma = 0.0385$, $a = -1.27$ and $b = -0.68$. Referring to the discussion in [13], one can choose $\bar{\delta} = a$ such that the inequality (5.10) holds. Then, one can choose $\varepsilon = 0.5$ such that the inequality (5.12) holds. That is, the hyper-network generating from the evolving algorithm with $m = 3$ and $N = 100$ can achieve synchronization from any initial values.

Figure 5.4 shows the orbits of the synchronization errors.

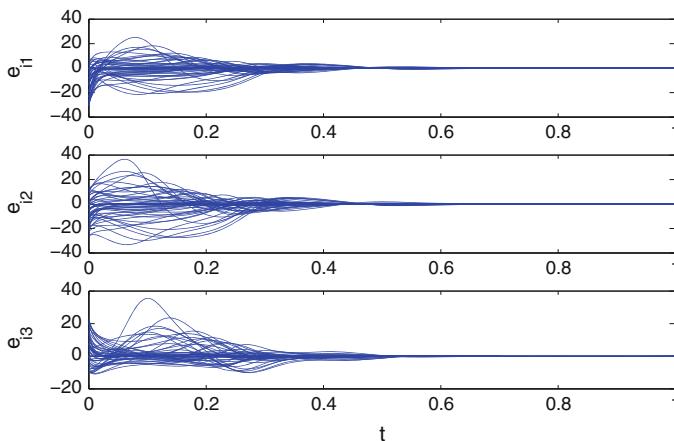


Fig. 5.4 Orbit of synchronization errors $e_{ij}(t)$, $i = 1, 2, \dots, 100$, $j = 1, 2, 3$

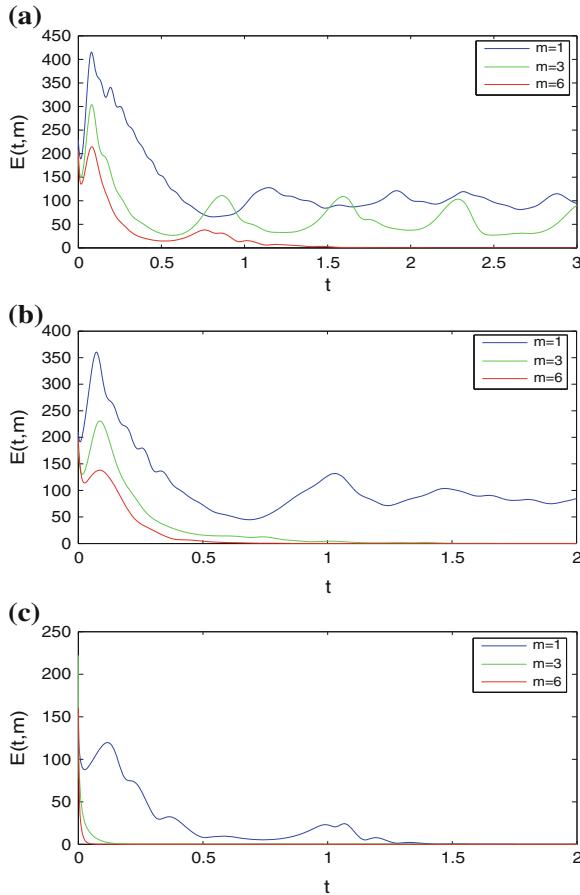


Fig. 5.5 Synchronization errors $E = \sqrt{\sum_{i=2}^N (x_i(t) - x_1(t))^T (x_i(t) - x_1(t))}$. Case (a) $\varepsilon = 0.3$, (b) $\varepsilon = 0.5$, (c) $\varepsilon = 5$

Next, consider the synchronizability of the 3-uniform hyper-network with different pairs of m and N .

It can be seen from Fig. 5.3 that $\lambda_2(N, 1) > \lambda_2(N, 2) > \dots > \lambda_2(N, 6)$, which means the hyper-network with a larger m has a stronger synchronizability.

Figure 5.5 shows the synchronization errors with $N = 100$ and different coupling strengths of $\varepsilon = 0.3$, $\varepsilon = 0.5$ and $\varepsilon = 5$, respectively, from which one can easily find that the hyper-network with $m = 6$ has the strongest synchronizability.

5.2.2 Synchronization on Nonlinearly-Coupled Hyper-Networks

Consider a p -uniform hyper-network consisting of N nodes with a nonlinear coupling, which is described by [15]

$$\dot{x}_i = F(x_i) + \sigma \sum_{1 \leq i_1 \leq i_2 \leq \dots \leq i_{p-1} \leq N} L_{ii_1i_2\dots i_{p-1}} Q(x_{i_1}, x_{i_2}, \dots, x_{i_{p-1}}), \quad (5.13)$$

where $i = 1, 2, \dots, N$, $x_i = (x_{i1}, x_{i2}, \dots, x_{in})^T$ is the state variable of node i , F is the n -dimensional nonlinear function describing the dynamics in each node, and σ is the coupling strength. In addition, L is the $\underbrace{N \times N \times \dots \times N}_{p \text{ times}}$ coupling array with elements $L_{i_1i_2\dots i_p} = 1$ if $i_1 \neq i_2 \neq \dots \neq i_p$, and there is a hyper-edge connecting the nodes i_1, i_2, \dots, i_p , $L_{ii\dots i} = -k_i$, where k_i is the hyper-degree of node i , and $L_{i_1i_2\dots i_p} = 0$ otherwise. Moreover, Q is the n -dimensional coupling function depending on the states of the oscillators in the $p-1$ nodes linked by a hyper-edge to node i , which is assumed to be invariant under any permutation of the node indices i_1, i_2, \dots, i_{p-1} . Thus, all hyper-edges correspond to identical interactions of groups of p connected oscillators, and these interactions involve all p oscillators within each hyper-edge in a symmetric manner.

Let $x_1 = x_2 = \dots = x_N \equiv x_s$ be the synchronization manifold, and $x_i = x_s + \xi_i$. Then, one has a linear approximation of Eq. (5.13) about x_s , as

$$\dot{\xi}_i = DF(x_i) \Big|_s \xi_i + \sigma \sum_{1 \leq i_1 \leq i_2 \leq \dots \leq i_{p-1} \leq N} L_{ii_1i_2\dots i_{p-1}} DQ(x_{i_1}, x_{i_2}, x_{i_1}, x_{i_{p-1}}) \Big|_s \xi, \quad (5.14)$$

where $DF(x_i) \Big|_s = DF(x_s)$ is the $n \times n$ Jacobian matrix of function F evaluated on the synchronization manifold. And $DQ(x_{i_1}, x_{i_2}, \dots, x_{i_{p-1}}) \Big|_s$ is the $n \times n(p-1)$ Jacobian matrix of the coupling function Q evaluated on the synchronization manifold, with $\xi = (\xi_{i_1}^T, \xi_{i_2}^T, \dots, \xi_{i_{p-1}}^T)^T$.

Let $\frac{\partial Q}{\partial x_{i_k}}(x_{i_1}, x_{i_2}, x_{i_1}, x_{i_{p-1}}) \Big|_s$ denote a sub-Jacobian of Q . Then,

$$DQ(x_{i_1}, x_{i_2}, x_{i_1}, x_{i_{p-1}}) \Big|_s \xi = \sum_{k=1}^{p-1} \frac{\partial Q}{\partial x_{i_k}}(x_{i_1}, x_{i_2}, \dots, x_{i_{p-1}}) \Big|_s \xi_{i_k}.$$

Since the coupling function $Q(x_{i_1}, x_{i_2}, x_{i_1}, x_{i_{p-1}})$ is invariant under any permutation of its arguments, it is possible to define another n -dimensional function, $\tilde{Q}(x_{i_k}) = \frac{1}{p-1} Q(x_{i_k}, x_{i_k}, \dots, x_{i_k})$ of only n variables $x_{i_k}^{(1)}, x_{i_k}^{(2)}, \dots, x_{i_k}^{(n)}$, which, on the synchronization manifold, satisfies the condition

$$D\tilde{Q}(x_{i_k}) \Big|_s = \frac{\partial Q}{\partial x_{i_k}}(x_{i_1}, x_{i_2}, \dots, x_{i_{p-1}}) \Big|_s,$$

where $D\tilde{Q}(x_{i_k}) \Big|_s = D\tilde{Q}(x_s)$ is the $n \times n$ Jacobian matrix of the function \tilde{Q} evaluated on the synchronization manifold.

Then, the Eq. (5.14) can be written as

$$\dot{\xi}_i = DF(x_i) \Big|_s \xi_i + \sigma \sum_{j=1}^N G_{ij} D\tilde{Q}(x_j)|_s \xi_j,$$

where G is the $N \times N$ symmetric matrix with diagonal elements $G_{ii} = -k_i(p-1)$ and off-diagonal elements $G_{ij} = \sum_{1 \leq i_1 \leq i_2 \leq \dots \leq i_{p-2} \leq N} L_{iji_1i_2\dots i_{p-2}}$ equal to the number of different hyper-edges linking the nodes i and j .

Then, the stability of the synchronization manifold can be investigated using the Master Stability Function method for weighted networks [16], as detailed in [15].

5.3 Synchronization on Colored Networks

A colored graph consists of colored nodes and colored edges, in which nodes with different colors mean that they have different properties and a pair of nodes connected by different color edges indicates that they have different relations [17–22].

A network corresponding to a colored graph is called a colored network, in which nodes with different colors means that they have different node-dynamics and a pair of nodes connected by different colored edges indicate that they have different mutual interactions. In particular, networks of coupled nonidentical dynamical systems with identical inner-coupling matrices can be regarded as node-colored networks, while networks of coupled identical dynamical systems with nonidentical inner-coupling matrices can be viewed as edge-colored networks [23].

For illustration, Fig. 5.6 shows a colored network consisting of 6 colored nodes and 9 colored edges.

Fig. 5.6 A colored network consisting of 6 colored nodes and 9 colored edges

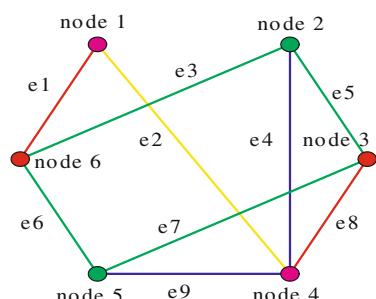
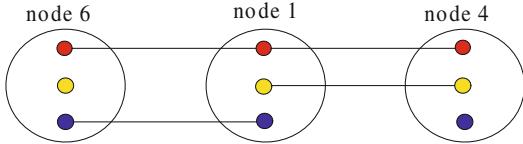


Fig. 5.7 (Color online) The red, yellow and blue points denote the first, second and third components of each individual node, respectively



5.3.1 Colored Networks with Same-Dimensional Node Dynamics

Consider a colored network consisting of N nodes with same-dimensional node dynamics, described by

$$\dot{x}_i(t) = f_i(x_i(t)) + \varepsilon \sum_{j=1, j \neq i}^N a_{ij} H_{ij}(x_j(t) - x_i(t)), \quad i = 1, 2, \dots, N, \quad (5.15)$$

where $x_i(t) = (x_{i1}(t), \dots, x_{in}(t))^T \in R^n$ is the state vector of node i , $f_i : R^n \rightarrow R^n$ is continuously differentiable, which represents the local dynamics of node i , and $\varepsilon > 0$ is the coupling strength. In addition, matrix $A = (a_{ij})_{N \times N}$ is the outer-coupling matrix representing the network topology, defined as follows: if there is a connection between node i and node j ($j \neq i$), then $a_{ij} > 0$; otherwise, $a_{ij} = 0$, and all the diagonal elements $a_{ii} = 0$. Moreover, $H_{ij} = \text{diag}(h_{ij}^1, h_{ij}^2, \dots, h_{ij}^n)$ is the inner-coupling matrix, which denotes the mutual interactions between nodes i and j , and is defined as follows: if the k th component of node i is affected by that of node j , then $h_{ij}^k \neq 0$; otherwise, $h_{ij}^k = 0$.

Figure 5.7 shows the detailed mutual interactions between node 1 and node 4 as well as between node 1 and node 6, with $H_{14} = \text{diag}(1, 1, 0)$ and $H_{16} = \text{diag}(1, 0, 1)$. That is, the first and second components of node 1 are affected by that of node 4, and the first and third components of node 1 are affected by that of node 6.

Let $c_{ij} = \text{diag}(c_{ij}^1, c_{ij}^2, \dots, c_{ij}^n)$, where $c_{ij}^k = a_{ij} h_{ij}^k$ for $i \neq j$ and $c_{ii}^k = -\sum_{j=1, j \neq i}^N c_{ij}^k$. Then, the colored network (5.15) becomes

$$\dot{x}_i(t) = f_i(x_i(t)) + \varepsilon \sum_{j=1}^N c_{ij} x_j(t), \quad i = 1, 2, \dots, N. \quad (5.16)$$

Let $C_k = (c_{ij}^k) \in R^{N \times N}$, $k = 1, 2, \dots, n$. In this sense, the colored network (5.16) can be regarded as a combination of n component subnetworks with a topology determined by C_k , $k = 1, 2, \dots, n$. The objective here is to ensure network (5.16) synchronize to a desired orbit $s(t)$, as discussed above.

Assumption 1 ([24]) Suppose that there exist positive definite matrices $\Delta_i = \text{diag}(\delta_{i1}, \delta_{i2}, \dots, \delta_{in})$ such that

$$(y - x)^T (f_i(y) - f_i(x)) \leq (y - x)^T \Delta_i (y - x)$$

for all $x, y \in R^n$, $i = 1, 2, \dots, N$.

Theorem 5.3 ([23]) Under Assumption 1, the following controlled edge-colored network (5.17) with adaptive feedback controllers (5.18)

$$\dot{x}_i(t) = f(x_i(t)) + \varepsilon \sum_{j=1}^N c_{ij} x_j(t) + u_i(t), \quad (5.17)$$

$$u_i(t) = -d(t)(x_i(t) - s(t)),$$

$$\dot{d}_k(t) = \eta_k \sum_{i=1}^N (x_{ik}(t) - s_k(t))^2, \quad (5.18)$$

can achieve synchronization, where $d(t) = \text{diag}(d_1(t), d_2(t), \dots, d_n(t))$ is an adaptive feedback control gain matrix and $\eta_k > 0$ are adaptive gains, $k = 1, 2, \dots, n$.

Example 5.2 ([23]) Consider an edge-colored network consisting of 100 coupled Lorenz systems [25],

$$\begin{pmatrix} \dot{x}_{i1} \\ \dot{x}_{i2} \\ \dot{x}_{i3} \end{pmatrix} = \begin{pmatrix} -10 & 10 & 0 \\ 28 & -1 & 0 \\ 0 & 0 & -8/3 \end{pmatrix} \begin{pmatrix} x_{i1} \\ x_{i2} \\ x_{i3} \end{pmatrix} + \begin{pmatrix} 0 \\ -x_{i1}x_{i3} \\ x_{i1}x_{i2} \end{pmatrix}.$$

In numerical simulations, the desired synchronized orbit is chosen as a chaotic orbit of the Lorenz system, generated with initial value $s(0) = (1, 2, 3)^T$. The topology of the three-component subnetworks is generated as a random graph [26] with $p = 0.1$, $p = 0.2$ and $p = 0.3$, respectively. To achieve synchronization, adaptive feedback controllers are applied. In the controlled edge-colored network (5.17) and the adaptive feedback controllers (5.18), the coupling strength is set as $\varepsilon = 1$, the initial value of $d(t)$ is $d(0) = \text{diag}(5, 5, 5)$, and the adaptive gain is $\eta_k = 0.05$. The initial values of the state variables are chosen randomly.

Figure 5.8 shows the orbits of the synchronization errors and $d_k(t)$.

Theorem 5.4 ([23]) Suppose that Assumption 1 holds and that the matrices C_k are irreducible and symmetric, and all of their off-diagonal entries are nonnegative. Then, the following controlled edge-colored network (5.19) with adaptive coupling strength and pinning controllers (5.20) can achieve network synchronization:

$$\dot{x}_i(t) = f(x_i(t)) + \varepsilon(t) \sum_{j=1}^N c_{ij} x_j(t) + u_i(t), \quad (5.19)$$

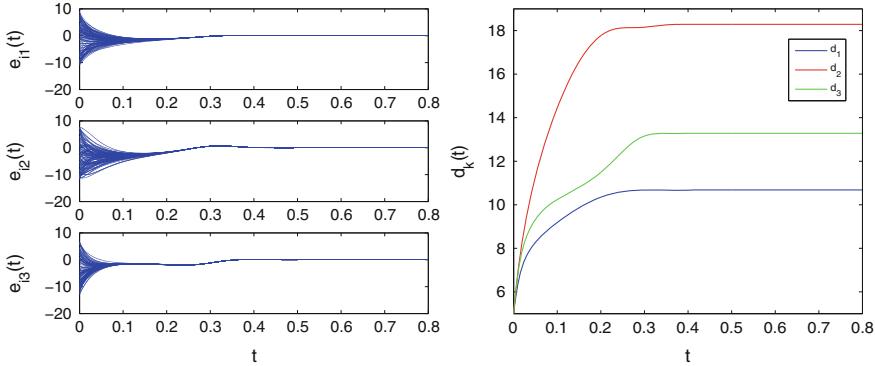


Fig. 5.8 *Left:* orbits of synchronization errors. *Right:* orbits of $d_k(t)$

$$\begin{aligned} u_i(t) &= -\varepsilon(t)d_i(x_i(t) - s(t)), \quad i = 1, 2, \dots, l, \\ u_i(t) &= 0, \quad \quad \quad i = l+1, \dots, N, \\ \dot{\varepsilon}(t) &= \gamma \sum_{i=1}^N (x_i(t) - s(t))^T (x_i(t) - s(t)), \end{aligned} \quad (5.20)$$

where $d_i = \text{diag}(d_{i1}, d_{i2}, \dots, d_{in})$, $i = 1, 2, \dots, l$, are positive definite feedback gain matrices, and $\gamma > 0$ is the adaptive gain.

Example 5.3 ([23]) Consider an edge-colored network consisting of 100 coupled Rössler systems [27]

$$\begin{pmatrix} \dot{x}_{i1} \\ \dot{x}_{i2} \\ \dot{x}_{i3} \end{pmatrix} = \begin{pmatrix} 0 & -1 & -1 \\ 1 & 0.2 & 0 \\ 0 & 0 & -5.7 \end{pmatrix} \begin{pmatrix} x_{i1} \\ x_{i2} \\ x_{i3} \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0.2 + x_{i1}x_{i3} \end{pmatrix}.$$

In numerical simulations, the first-component subnetwork is generated as the BA scale-free network [6] with $m_0 = m = 3$, the second is generated as the NW small-world network [28] with $K = 4$ and $p = 0.1$, and the third is generated as the WS small-world network [29] with $K = 6$ and $p = 0.1$. So they are very different.

In the controlled edge-colored network (5.19) and the controllers (5.20), the feedback gain matrix is chosen as $d_i = \text{diag}(10, 10, 10)$ for $i = 1, 2, \dots, 10$, the initial value of $\varepsilon(t)$ is $\varepsilon(0) = 5$, and the adaptive gain is $\gamma = 0.01$. The initial values of the state variables are chosen randomly.

Figure 5.9 shows the orbits of the synchronization errors and $\varepsilon(t)$.

Theorem 5.5 ([23]) Suppose that Assumption 1 holds and that the matrices C_k are irreducible and symmetric, and all of their off-diagonal entries are nonnegative.

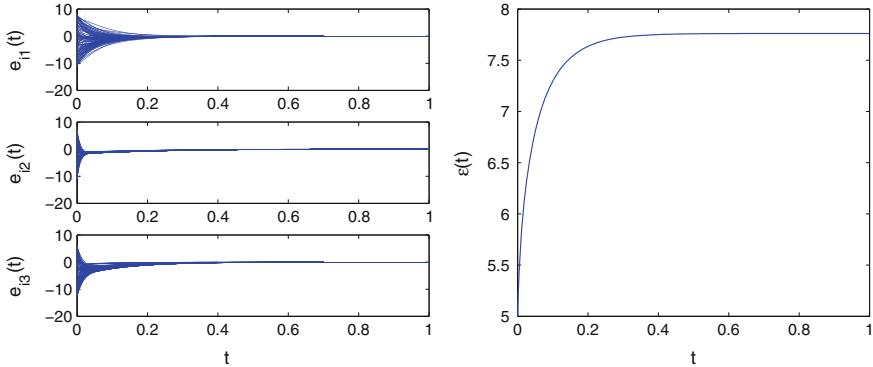


Fig. 5.9 *Left:* orbits of synchronization errors. *Right:* orbits of $\varepsilon(t)$

Then, the following controlled colored network (5.21) with adaptive coupling strength and open-plus-close-loop controllers (5.22) can achieve network synchronization:

$$\dot{x}_i(t) = f(x_i(t)) + \varepsilon(t) \sum_{j=1}^N c_{ij} x_j(t) + u_i(t), \quad (5.21)$$

$$\begin{aligned} u_i(t) &= u_i^o(t) + u_i^c(t), \\ u_i^o(t) &= \dot{s}(t) - f_i(s(t)), \\ u_i^c(t) &= -\varepsilon(t) d_i(x_i(t) - s(t)), \quad i = 1, 2, \dots, l, \\ u_i^c(t) &= 0, \quad \quad \quad i = l+1, \dots, N, \\ \dot{\varepsilon}(t) &= \gamma \sum_{i=1}^N (x_i(t) - s(t))^T (x_i(t) - s(t)), \end{aligned} \quad (5.22)$$

where $u_i^o(t)$ and $u_i^c(t)$ are open- and close-loop controllers respectively, $d_i = \text{diag}(d_{i1}, d_{i2}, \dots, d_{in})$, $i = 1, 2, \dots, l$, are positive definite feedback gain matrices, and $\gamma > 0$ is the adaptive gain.

Example 5.4 ([23]) Consider a colored network (5.19) consisting of 50 Lorenz systems and 50 Rössler systems. The desired synchronized orbit is chosen as the trajectory of the Chen system [30]:

$$\begin{pmatrix} \dot{s}_1 \\ \dot{s}_2 \\ \dot{s}_3 \end{pmatrix} = \begin{pmatrix} -35 & 35 & 0 \\ -7 & 28 & 0 \\ 0 & 0 & -3 \end{pmatrix} \begin{pmatrix} s_1 \\ s_2 \\ s_3 \end{pmatrix} + \begin{pmatrix} 0 \\ -s_1 s_3 \\ s_1 s_2 \end{pmatrix},$$

with initial values $s(0) = (1, 2, 3)^T$. Clearly, these three chaotic systems are quite different.

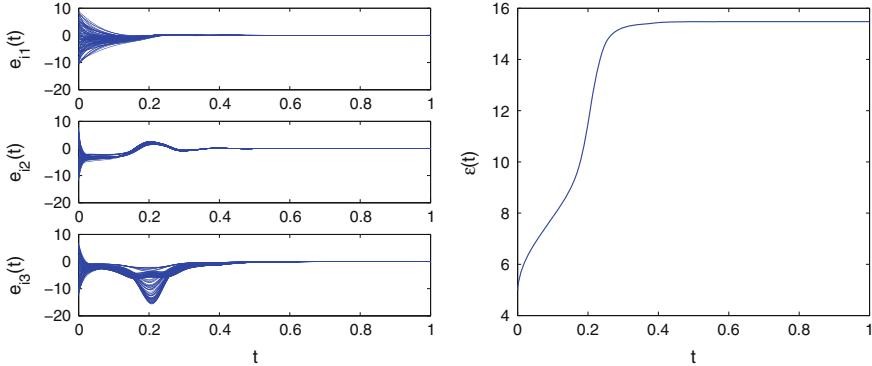


Fig. 5.10 *Left:* orbits of synchronization errors. *Right:* orbits of $\varepsilon(t)$

In numerical simulations, the same three-component subnetworks are generated and the same parameters are chosen as in the above Example 5.2. Figure 5.10 shows the orbits of the synchronization errors and $\varepsilon(t)$.

5.3.2 Colored Networks with Different-Dimensional Node Dynamics

Consider a general colored network consisting of N nodes with different-dimensional node dynamics, described by [31]

$$\dot{x}_i(t) = f_i(x_i(t)) + \varepsilon \sum_{j=1, j \neq i}^N a_{ij} \left(H_{ij} x_j(t) - H_{ii}^j x_i(t) \right), \quad (5.23)$$

where $i = 1, 2, \dots, N$, $x_i \in R^{n_i}$ is the state variable of node i , $f_i : R^{n_i} \rightarrow R^{n_i}$ denotes the local dynamics of node i , and $\varepsilon > 0$ is the coupling strength. Moreover, $A = (a_{ij})_{N \times N}$ is the outer-coupling matrix: if there is a connection between node i and node j ($j \neq i$), then $a_{ij} \neq 0$; otherwise, $a_{ij} = 0$. Also, $H_{ii}^j = \text{diag}(h_{ii}^{j(1)}, h_{ii}^{j(2)}, \dots, h_{ii}^{j(n_i)}) \in R^{n_i \times n_i}$ and $H_{ij} = (h_{ij}^{(kl)})_{n_i \times n_j} \in R^{n_i \times n_j}$ ($i \neq j$) are inner-coupling matrices, which denote the mutual interactions between nodes i and j , defined as follows: if the k th component of node i is affected by the l th component of node j , then $h_{ii}^{j(k)} = 1$ and $h_{ij}^{(kl)} = 1$; otherwise, they are zero.

Figure 5.11 shows in detail the mutual interactions between node 6 and 1 as well as between node 6 and node 2, with

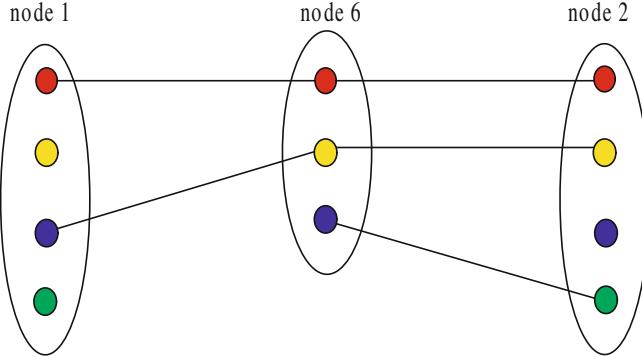


Fig. 5.11 (Color online) The *red*, *yellow*, *blue* and *green* points denote the first, second, third and fourth components of each individual node, respectively

$$H_{66}^1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad H_{61} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \quad H_{66}^2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad H_{62} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Definition 5.1 ([31]) Let $s(t) \in R^m$ be a solution of a dynamical system $\dot{s}(t) = g(s(t))$. The generalized matrix projective synchronization of network (5.23) is said to be achieved, if there exist N matrices $F_i \in R^{n_i \times m}$, $i = 1, 2, \dots, N$, such that

$$\lim_{t \rightarrow \infty} \|x_i(t) - F_i s(t)\| = 0, \quad i = 1, 2, \dots, N.$$

Accordingly, some well-known notions of network synchronization, such as complete synchronization, projective synchronization, reduced or increased-order synchronization, etc., are special cases of Definition 5.1. For example, if $n_1 = n_2 = \dots = n_N = m$ and F_i is the identity matrix in network (5.23), then the network becomes a network consisting of different nodes with the same dimension; when complete synchronization is considered, it is the case being studied earlier, e.g., in [24].

Assumption 2 Suppose that there exist N constant diagonal matrices, $\Delta_i \in R^{n_i \times n_i}$, $i = 1, 2, \dots, N$, such that the vector functions $f_i(x_i(t))$ satisfy

$$(y_i - x_i)^T (f_i(y_i) - f_i(x_i)) \leq (y_i - x_i)^T \Delta_i (y_i - x_i), \quad \forall x_i, y_i \in R^{n_i}.$$

Theorem 5.6 ([31]) Under Assumption 2, generalized matrix projective synchronization of the controlled network (5.24), with adaptive open-plus-closed-loop controllers and updating laws (5.25), can be achieved:

$$\dot{x}_i(t) = f_i(x_i(t)) + \varepsilon \sum_{j=1, j \neq i}^N a_{ij} (H_{ij}x_j(t) - H_{ii}^j x_i(t)) + u_i(t), \quad (5.24)$$

$$\begin{aligned} u_i(t) &= u_i^o(t) + u_i^c(t), \\ u_i^o(t) &= F_i \dot{s}(t) - f_i(F_i s(t)) - \varepsilon \sum_{j=1, j \neq i}^N a_{ij} (H_{ij} F_j s(t) - H_{ii}^j F_i s(t)), \\ u_i^c(t) &= -d_i(t)(x_i(t) - F_i s(t)), \\ \dot{d}_i(t) &= \theta_i(x_i(t) - F_i s(t))^T (x_i(t) - F_i s(t)), \end{aligned} \quad (5.25)$$

where $i = 1, 2, \dots, N$, and $\theta_i > 0$ are adaptive gains.

Theorem 5.7 ([31]) Suppose that Assumption 2 holds. Choose the open-plus-closed-loop controllers in network (5.24) as follows:

$$\begin{aligned} u_i^o(t) &= F_i \dot{s}(t) - f_i(F_i s(t)) - \varepsilon \sum_{j=1, j \neq i}^N a_{ij} (H_{ij} F_j s(t) - H_{ii}^j F_i s(t)), \\ u_i^c(t) &= -K_i(t)(x_i(t) - F_i s(t)), \\ K_i(t) &= \begin{cases} K_i, & t \in [mT, mT + \delta], \\ 0, & t \in [mT + \delta, (m+1)T], \end{cases} \end{aligned} \quad (5.26)$$

where $i = 1, 2, \dots, N$, $m = 0, 1, 2, \dots$, $K_i > 0$ are the feedback control gains, $T > 0$ is the control period, and $\delta \in (0, T)$ is the control duration. Then, generalized matrix projective synchronization of network (5.24) can be achieved, if there exist positive constants c_1 and c_2 such that

$$\begin{aligned} (i) \quad &\Omega + \varepsilon G + c_1 I - \widehat{K} \leq 0, \\ (ii) \quad &\Omega + \varepsilon G - c_2 I \leq 0, \\ (iii) \quad &c_2 - (c_1 + c_2)\theta < 0, \end{aligned} \quad (5.27)$$

where $\theta = \delta/T$, $\widehat{K} = \text{diag}(K_1 I_{n_1}, K_2 I_{n_2}, \dots, K_N I_{n_N})$, and I is an identity matrix of appropriate dimension.

Example 5.5 ([31]) Consider the general colored network consisting of 6 nodes and 9 edges shown in Fig. 5.2, with nodes 1 and 4 being the hyperchaotic Lorenz system: [32]

$$\begin{aligned} \dot{x}_1 &= 10(x_2 - x_1) + x_4, \\ \dot{x}_2 &= 28x_1 - x_2 - x_1 x_3, \\ \dot{x}_3 &= x_1 x_2 - 8/3 x_3, \\ \dot{x}_4 &= 1.3x_4 - x_1 x_3, \end{aligned}$$

nodes 2 and 5 being the hyperchaotic Lü system [33]:

$$\begin{aligned}\dot{y}_1 &= 36(y_2 - y_1) + y_4, \\ \dot{y}_2 &= 20y_2 - y_1y_3, \\ \dot{y}_3 &= y_1y_2 - 3y_3, \\ \dot{y}_4 &= y_1y_3 + y_4,\end{aligned}$$

and nodes 3 and 6 being the Chua circuit [34]:

$$\begin{aligned}\dot{z}_1 &= 9(z_2 - z_1 - \varphi(z_1)), \\ \dot{z}_2 &= z_1 - z_2 + z_3, \\ \dot{z}_3 &= -100/7z_2,\end{aligned}$$

where $\varphi(z_1) = -5/7z_1 - 3/14(|z_1 + 1| - |z_1 - 1|)$. Clearly, they are very different dynamics.

Choose the signal $s(t)$ in Definition 5.1 as a solution of the Chen system [30]:

$$\begin{aligned}\dot{s}_1 &= 35(s_2 - s_1), \\ \dot{s}_2 &= 28s_2 - 7s_1 - s_1s_3, \\ \dot{s}_3 &= s_1s_2 - 3s_3,\end{aligned}$$

with initial values $s(0) = (1, 2, 3)^T$, and F_i ($i = 1, 2, \dots, 6$) being

$$F_1 = F_4 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}, \quad F_2 = F_5 = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix},$$

and $F_3 = F_6 = I_3$, where I_3 is the 3×3 identity matrix.

Firstly, consider generalized matrix projective synchronization of the controlled network (5.24) with the adaptive open-plus-close-loop controllers (5.25).

In numerical simulations, for $i = 1, 2, \dots, 6$, choose the initial values of $d_i(t)$ as $d_i(0) = 0.5$, adaptive gain $\theta_i = 0.01$, coupling strength $\varepsilon = 2$, and initial values of state variables $x_i(t)$ at random. Figures 5.12 and 5.13 show the orbits of synchronization errors and adaptive feedback gains $d_i(t)$, respectively.

Secondly, consider generalized matrix projective synchronization of the controlled network (5.24) with the open-plus-close-loop controllers (5.26). In numerical simulations, choose $\varepsilon = 0.02$, $T = 0.1$, $\theta = 0.75$, and the initial values of state variables $x_i(t)$ randomly.

Figure 5.14 shows the orbits of synchronization errors.

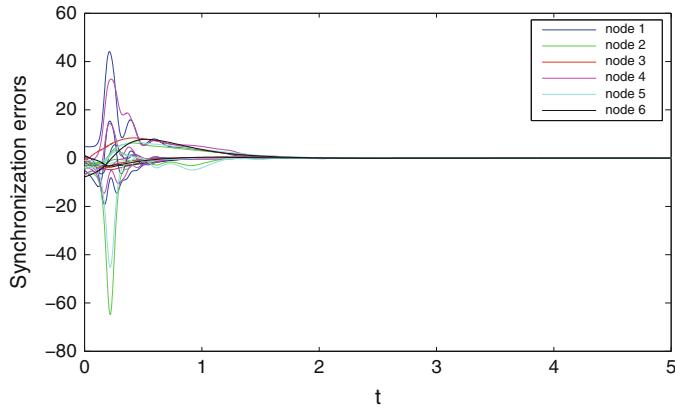


Fig. 5.12 Orbitas de errores de sincronización $e_i(t) = x_i(t) - F_i s(t)$, $i = 1, 2, \dots, 6$

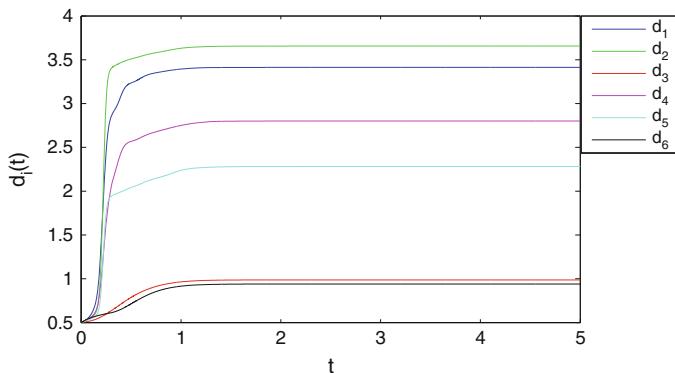


Fig. 5.13 Orbitas de $d_i(t)$, $i = 1, 2, \dots, 6$

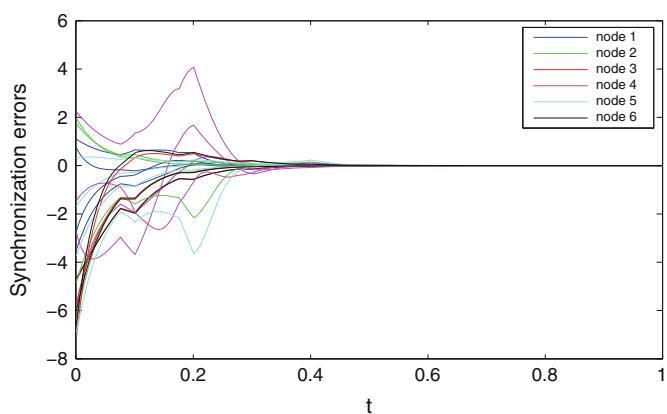


Fig. 5.14 Orbitas de errores de sincronización $e_i(t) = x_i(t) - F_i s(t)$, $i = 1, 2, \dots, 6$

References

1. Berge, C.: Graphs and Hypergraphs, vol. 6. Elsevier, New York (1973)
2. Berge, C.: Hypergraphs: Combinatorics of Finite Sets, Vol. 45. North-Holland Holl, Amsterdam (1989)
3. Yang, G.Y., Hu, Z.L., Liu, J.G.: Knowledge diffusion in the collaboration hypernetwork. *Physica A* **419**, 429–436 (2015)
4. Hu, F., Zhao, H.X., He, J.B., Li, F.X., Li, S.L., Zhang, Z.K.: An evolving model for hypergraph-structure-based scientific collaboration networks. *Acta Phys. Sin.* **62**, 198901 (2013) (in Chinese)
5. Wang, J.W., Rong, L.L., Deng, Q.H., Zhang, J.Y.: Evolving hypernetwork model. *Eur. Phys. J. B* **77**, 493–498 (2010)
6. Barabási, A.-L., Albert, R.: Emergence of scaling in random networks. *Science* **286**, 509–512 (1999)
7. Wu, Z.Y., Duan, J.Q., Fu, X.C.: Synchronization of an evolving complex hyper-network. *Appl. Math. Model.* **38**, 2961–2968 (2014)
8. Guo, J.L., Zhu, X.Y.: Emergence of scaling in hypernetworks. *Acta Phys. Sin.* **63**, 090207 (2014) (in Chinese)
9. Guo, J.L.: Emergence of scaling in non-uniform hypernetworks—does the rich get richer lead to a power-law distribution? *Acta Phys. Sin.* **63**, 208901 (2014) (in Chinese)
10. Yang, G.Y., Liu, J.G.: A local-world evolving hypernetwork model. *Chin. Phys. B* **23**, 018901 (2014)
11. Hu, F., Zhao, H.X., Ma, X.J.: An evolving hypernetwork model and its properties. *Sin. China. Phys. Mech. Astron.* **43**, 16–22 (2013) (in Chinese)
12. Chen, G.R., Wang, X.F., Li, X.: Introduction to Complex Networks: Models, Structure and Dynamics. High Education Press, Beijing (2012)
13. Wang, X.F., Chen, G.R.: Synchronization in scale-free dynamical networks: robustness and fragility. *IEEE Trans. Circuits Syst.* **I** (49), 54–62 (2002)
14. Chua, L.O., Wu, C.W., Huang, A., Zhong, G.Q.: A universal circuit for studying and generating chaos—part I: Routes to chaos. *IEEE Trans. Circuits Syst.* **I** (40), 732–742 (1993)
15. Krawiecki, A.: Chaotic synchronization on complex hypergraphs. *Chaos Solitons and Fractals* **65**, 44–50 (2014)
16. Chavez, M., Hwang, D.-U., Amann, A., Hentschel, H.G.E., Boccaletti, S.: Synchronization is enhanced in weighted complex networks. *Phys. Rev. Lett.* **94**, 218701 (2005)
17. Hell, P., Manoussakis, Y., Tuza, Z.: Packing problems in edge-colored graphs. *Discrete Appl. Math.* **52**, 295–306 (1994)
18. Zielonka, W.: Infinite games on finitely coloured graphs with applications to automata on infinite trees. *Theor. Comput. Sci.* **200**, 135–183 (1998)
19. Fujita, S., Nakamigawa, T.: Balanced decomposition of a vertex-colored graph. *Discrete Appl. Math.* **156**, 3339–3344 (2008)
20. Becu, J.M., Dah, M., Manoussakis, Y., Mendy, G.: Links in edge-colored graphs. *Eur. J. Combin.* **31**, 442–460 (2010)
21. Orlitsky, A., Venkatesh, S.S.: On edge-colored interior planar graphs on a circle and the expected number of RNA secondary structures. *Discrete Appl. Math.* **64**, 151–178 (1996)
22. Wang, Y., Desmedt, Y.: Edge-colored graphs with applications to homogeneous faults. *Inform. Process. Lett.* **111**, 634–641 (2011)
23. Wu, Z.Y., Xu, X.J., Chen, G.R., Fu, X.C.: Adaptive synchronization and pinning control of colored networks. *Chaos* **22**, 043137 (2012)
24. Song, Q., Cao, J., Liu, F.: Synchronization of complex dynamical networks with nonidentical nodes. *Phys. Lett. A* **374**, 544–551 (2010)
25. Lorenz, E.: Deterministic nonperiodic flow. *J. Atmos. Sci.* **20**, 130–141 (1963)
26. Erdős, P., Rényi, A.: On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci.* **5**, 17–60 (1960)

27. Rössler, O.E.: An equation for continuous chaos. *Phys. Lett. A* **57**, 397–398 (1976)
28. Newman, M.E.J., Watts, D.J.: Renormalization group analysis of the small-world network model. *Phys. Lett. A* **263**, 341–346 (1999)
29. Watts, D.J., Strogatz, S.H.: Collective dynamics of ‘small-world’ networks. *Nature* **393**, 440–442 (1998)
30. Chen, G.R., Ueta, T.: Yet another chaotic attractor. *Int. J. Bifur. Chaos* **9**, 1465–1466 (1999)
31. Wu, Z.Y., Xu, X.J., Chen, G.R., Fu, X.C.: Generalized matrix projective synchronization of general colored networks with different-dimensional node dynamics. *J. Franklin Inst.* **351**, 4584–4595 (2014)
32. Jia, Q.: Hyperchaos generated from the Lorenz chaotic system and its control. *Phys. Lett. A* **366**, 217–222 (2007)
33. Chen, A., Lu, J., Lü, J., Yu, S.: Generating hyperchaotic Lü attractor via state feedback control. *Physica A* **364**, 103–110 (2006)
34. Belykh, V.N., Chua, L.O.: New type of strange attractor from a geometric model of Chua’s circuit. *Int. J. Bifur. Chaos* **2**, 697–704 (1992)

Chapter 6

New Nonlinear CPRNG Based on Tent and Logistic Maps

Oleg Garasym, Ina Taralova and René Lozi

Abstract This paper is devoted to the design of new chaotic Pseudo Random Number Generator (CPRNG). Exploring several topologies of network of 1-D coupled chaotic mapping, we focus first on two dimensional networks. Two coupled maps are studied: TTL^{RC} non-alternative, and TTL^{SC} alternative. The primary idea of the novel maps has been based on an original coupling of the tent and logistic maps to achieve excellent random properties and homogeneous/uniform/density in the phase plane, thus guaranteeing maximum security when used for chaos base cryptography. In this aim a new nonlinear CPRNG: $MTTL_2^{SC}$ is proposed. In addition, we explore higher dimension and the proposed ring coupling with injection mechanism enables to achieve the strongest security requirements.

6.1 Introduction

The tremendous development of new IT technologies, e-banking, e-purchasing, etc. nowadays increases incessantly the needs for new and more secure cryptosystems. The latter are used for information encryption, pushing forward the demand for more efficient and secure pseudo-random number generators [29]. At the same time, chaotic maps show up as perfect candidates able to generate independent and secure pseudo-random sequences (used as information carriers or directly involved in the process of encryption/decryption). However, the majority of well-known chaotic maps are not naturally suitable for encryption [19] and most of them don't exhibit even satisfactory properties for encryption. To deal with this open problem, we propose the

O. Garasym · I. Taralova
IRCCyN UMR CNRS 6597, Ecole Centrale de Nantes, 44300 Nantes, France
e-mail: Oleg.garasym@ircyn.ec-nantes.fr

I. Taralova
e-mail: Ina.taralova@ircyn.ec-nantes.fr

R. Lozi (✉)
Laboratoire J. A. Dieudonné, UMR CNRS 7351, Université de Nice Sophia-Antipolis,
Parc Valrose, 06102 Nice, France
e-mail: rlozi@unice.fr

revolutionary idea to couple tent and logistic map, and to add an injection mechanism to bound the escaping orbits. Good results are demonstrated with two different kinds of coupling, simple and ring-coupling in dimension 2, thus increasing the complexity of the system. However as those results are not completely satisfactory, an improved geometry of coupling is introduced allowing us to describe a new 2-D Chaotic Pseudo Random Number Generator (CPRNG).

The various choice of the PRNG and crypto algorithms is necessary to provide reliable security system. We describe a software approach because it is easy to change cryptosystem to support protection whereas hardware requires more time and big expenses. For instance, after the secure software application called Wi-Fi Protected Access (WPA) protocol have been broken it was simply updated and no expensive hardware needed to be bought.

In the history, there are periods of popular algorithms, cryptographic methods and approaches until the moment they have been broken. It is a usual thing in information security one algorithm replaces another because information technologies and mathematics make progress. The one of today's open mathematical problem is factoring the product of two large prime numbers which is foundation for RSA algorithm. The RSA was created by Ron Rivest, Adi Shamir and Leonard Adleman in 1977 and since that time was implemented in widespread applications, is used for an independent cryptographic production [49]. Open coding based on the RSA algorithm is utilized in popular encryption package PGP, operating system Windows, various Internet browsers, banking computer systems. Moreover, there exist various international standards for public key cryptography and digital signatures. However, we expect the new cryptographical standards soon, because there are several evidences of weakness of those methods. Recently was known about a "back-door" in PRNG (Dual EC DRBG) which is implemented in RSA algorithm by default.

The second reason is that modern mathematical technology could give a possibility to break the process of obtaining cryptographic keys. In addition, there are many hackers attacks to RSA encryption, thus it could be broken in the nearest future.

Moreover, there is another important problem to be solved. RSA is a public keys system that is much simpler than a system with private keys such as: René Lozi and Estelle Cherrier [24], Safwan El Assad [32], Singh Ajit and Gilhotra Rimple [39].

Consequently, it is necessary to have an alternative way of secure information transmission. Chaos based methods are very promising for application in information security. One of the evidences is that needs for data protection are increased and encryption procedures requires to generate pseudo-random sequences with very long periods. The chaotic maps when used in stirling way could generate not only chaotic number but also pseudo-random numbers as we will show here.

Methods of nonlinear dynamics allow to create with relatively little effort a fundamentally new type of behavior, capable of holding, encrypting and process given information. Foundations of it are that chaotic attractors could contain an infinite set of unstable periodic behaviors. Nowadays there are different ways of chaos application to design symmetric and asymmetric cryptosystems. The methods based on circuits synchronization have been applied to numerous chaotic systems [33, 42]. Nonlinear dynamics is a promising direction to solve the problem of information

processing and organization of secure information transmission through the use of systems exhibiting chaotic dynamics.

Here we represent an original idea combining of tent and logistic maps for new chaotic PRNG design. Since, it is a very responsible and challenging task to design CPRNG applicable to cryptography, numerous analysis have been fulfilled. Essentially we focus on 2-D map as a more difficult task achieving excellent chaotic and randomness properties. The 3-steps injection mechanism, ring- and auto-coupling techniques are used to achieve complex and uniform dynamics. We demonstrate excellently puzzled chaotic dynamics in the space exhibiting sufficient randomness properties only for 2-D map. The most significant tests were successfully passed. Moreover, higher dimensional system here proposed as well. The systems provide also good candidates for CPRNG.

6.2 CPRNG Indistinguishable from Random

Let us consider a CPRNG that produces binary bits $G : K \rightarrow \{0, 1\}^n$, where K belongs to keyspace \mathbb{A} , n is a number of bits. In real life, for any given $K \in \mathbb{A}$ intruder should not distinguish it from random. Usually, statistical tests are used to the binary sequence analysis. The results could disclose some weakness in generated random sequences or at least refuse the truly random nature of the generator. Whereas statistical tests prove the behavior of the generator as being like truly random, which implies robustness against attacks based on such kind of analysis.

The statistical test is an algorithm that takes its inputs and as an output gives 0 or 1. The given sequence is supposed to be not random whereas output equals 0. In opposite case, where output equals 1 we assume that the given input is random, according to the test. All statistical tests are used to determine either the given sequence that produced by some generator $G(K)$ looks random or it does not look random. However, the well-known fact is when statistical test could make the wrong decision relatively to the sequence [35]. Therefore, it is preferable to define PRNG advantage [14].

The generator G exhibits weakness if the statistical test Υ was able to distinguish the output from random. However, if the advantage is close to zero $Adv \neq 0$ then the pseudo-random inputs have the same behavior as truly random within statistical test Υ . Therefore, Υ could not distinguish the generator from random.

Generator $G : K \rightarrow \{0, 1\}^n$ is a secure PRNG if for every efficient statistical tests $\Upsilon : Adv_{PRNG}[\Upsilon, G]$ is negligible or the statistical tests cannot distinguish its output from random.

There are hundreds of statistical tests that confirm or refuse randomness [31]. When for all the tests a given string looks like random, the generator is considered as robust. Due to the tests it will not be able to use statistical attacks on the algorithm if intruder cannot distinguish PRNG from truly random.

Chaotic functions deal with floating points, thus statistical tests are not directly efficient to define if a CPRNG is robust or not. The laws and standards for binary strings cannot fully guarantee robustness since the nature is different. Chaotic PRNG

can be used in appearance of noise e.g. in the CSK, CMA, CMI models or as binary string in *XOR*-function. Therefore, there are more requirements to CPRNG to prove indistinguishability from truly random generator. Note that, today there is no standards on CPRNG analysis, but the primary tests are described in the next section.

6.2.1 Some Tests for Robustness

There are no standards of security verification, making it difficult to determine whether the system is truly secure. This is the crucial reason why chaos is till not officially used in cryptography. However we represent the main and the most important security tests to chaotic systems.

Progress has been made to the point that chaos can be applied to secure communication [10, 48] and many papers focused on robust chaotic generator design [3, 7, 16]. There are several criteria respected by the community to the chaotic generators: Largest Lyapunov exponent [38], Chaotic attractor in the phase space [8, 18], phase delay [20, 26], Topological mixing [43, 46], Reactivity to small changes in initial conditions (chaotic sensitivity) [4, 41], Uniform distribution [6, 15], Autocorrelation [11, 15], Crosscorrelation [13], NIST tests [37, 44].

To prove robustness and ability to cryptosystems applications the chaotic system should demonstrate excellent randomness and chaoticity results. Even if there is no exact and unique chaos definition, the system is considered to be chaotic and could be applied to cryptosystems when the chaotic generator behaves as a performed PRNG (generated sequences must all be unitarily independent etc.). Therefore the following requirements should be fulfilled [1]:

- Random pattern: passes statistical tests of randomness;
- Long period: goes as long as possible before repeating;
- Efficiency: executes rapidly and requires little storage;
- Repeatability: produces same sequence if started with same initial conditions;
- Portability: runs on different kinds of computers and is capable of producing same sequence on each.

Therefore, for chaotic PRNG we use the following test: Largest Lyapunov exponent, autocorrelation, cross-correlation, test for uniform distribution, chaotic attractor in phase space and phase delay and finally NIST tests.

6.2.2 Uniform Distribution

Randomness is often associated with unpredictability. However, it is difficult to say if a sequence is predictable or no, thus it is considered as unpredictable if each of the points on the range interval has equal chance to be chosen. The test of uniform distribution gives the answer about probability of the points choice. If all points have

equal probability then the chance to predict the next point is very small. Thus, this test is important to analyse whether the sequence is unpredictable.

An excellent PRNG looks like truly random, means unpredictable or there are any correlation between points that have equal chance to be chosen. If the generator is capable to produce the sequences uniformly distributed in phase space and phase delay then the system behavior is like truly random.

There are different tools to analyse points distribution i.e. histogram, cumulative distribution. However they give very general information. In order to assess numerical computations more accurately and to qualitatively study the chaotic systems an approximation density function [21] is preferable. The approximation $P_{M,N}(x)$ is defined of the invariant measure (the probability distribution function) linked to the 1-dimensional map f going from the interval $J \subset \mathbb{R}$ into itself, when computed with floating numbers. The regular partition of M small intervals (boxes) r_i of J is defined by

$$s_i = -1 + \frac{2i}{M}, i = 0, M \quad (6.1)$$

$$r_i = [s_i, s_{i+1}[\quad i = 0, M-2 \text{ and } r_{M-1} = [s_{M-1}, 1] \quad (6.2)$$

the length of each box is equal to $2/M$ and the r_i intervals form a partition of the interval J

$$J = \bigcup_0^{M-1} r_i \quad (6.3)$$

All iterates $f^{(n)}(x)$ belonging to these boxes are collected, after a transient regime of Q iterations decided *a priori*, (i.e. the first Q iterates are neglected). Once the computation of $N + Q$ iterates is completed, the relative number of iterates with respect to N/M in each box r_i represents the value $P_N(s_i)$. The approximated $P_N(x)$ defined is then a step function, with M steps. As M may vary, it is defined by

$$P_{M,N}(s_i) = \frac{M}{N}(\#r_i) \quad (6.4)$$

where $\#r_i$ is the number of iterates belonging to the interval r_i . $P_{M,N}(x)$ is normalized to 2 on the interval $J = [-1, 1]$.

$$P_{M,N}(x) = P_{M,N}(s_i), \forall x \in r_i \quad (6.5)$$

If the chaotic system is combined of p -coupled maps, then it is important to analyse distribution of each component $x^1, x^2, x_1^2, \dots, x^p$ of X and variable X itself in J^p as well. The approximated probability distribution function, $P_{M,N}(x^j)$ associated to one among several components of $F(X)$. It is used equally N_{disc} for M and N_{iter} for N , when they are more explicit.

The discrepancies E_1 (in norm L_1), E_2 (in norm L_2) and E_∞ (in norm L_∞) between $P_{N_{disc}, N_{iter}}(x^j)$ and the Lebesgue measure, which is the invariant measure are defined by

$$E_{1, N_{disc}, N_{iter}}(x^j) = \|P_{N_{disc}, N_{iter}}(x^j) - 1\|_{L_1} \quad (6.6)$$

$$E_{2, N_{disc}, N_{iter}}(x^j) = \|P_{N_{disc}, N_{iter}}(x^j) - 1\|_{L_2} \quad (6.7)$$

$$E_{\infty, N_{disc}, N_{iter}}(x^j) = \|P_{N_{disc}, N_{iter}}(x^j) - 1\|_{L_\infty} \quad (6.8)$$

The numerical calculation of the uniform distribution allows us to judge about system unpredictability.

6.2.3 NIST Tests

Currently, NIST (National Institute of Standard and Technology) tests are the most powerful and widely used tool to test the sequences for randomness [37]. The standard includes 15 tests which on output give 188 results. The methodology allows with high probability to make conclusion about existing randomness in the sequences. According to the NIST tests the sequences are analysed as follow:

- 1 Zero hypothesis H_0 is putting forward. The assumption that the given binary sequence is random.
- 2 Statistic is calculated.
- 3 The probability value $P \in [0, 1]$ is calculated.
- 4 The probability value P is compared with significance level α , $\alpha \in [0.001; 0.01]$. If $P \geq \alpha$ then the hypothesis is accepted, otherwise another hypothesis is taken.

The results of the tested sequence take form of probability vector $P = \{P_1, P_2, \dots, P_{188}\}$. The P_i test indicates the weakness of the sequence. The standard recommends the sequence of 100 blocks per 10^6 bits. Thus, the sequence length should be equal to 10^8 bits. Each of the given 100 blocks passes the analysis. The testing results are consolidated to the summarised table when in front of the each test there is for example the value 97/100, that means that 97 is the number of blocks that successfully passed the test out of 100. The threshold of fail blocks are 3.

6.3 Exploring Topologies of Network of Coupled Chaotic Maps

In 1973, sir Robert May, a famous biologist introduced the nonlinear, discrete time dynamical system called logistic equation:

$$x_{n+1} = rx_n(1 - x_n) \quad (6.9)$$

as a model for the fluctuations in the population of fruit flies in a closed container with constant food [27]. Since that early time this logistic equation has been extensively studied especially by May [28], and Mitchell Feigenbaum [9] under the equivalent form:

$$x_{n+1} = f_\mu(x_n) \quad (6.10)$$

where

$$f_\mu(x_n) \equiv L_\mu(x) = 1 - \mu x^2 \quad (6.11)$$

Another often studied discrete dynamical system is defined by the symmetric tent map:

$$f_\mu(x_n) \equiv T_\mu(x_n) = 1 - \mu|x| \quad (6.12)$$

In both cases, μ is a control parameter that has impact to chaotic degree, and those mappings are sending the one-dimensional interval $[-1, 1]$ into itself.

Those two maps have also been fully explored in the hope of generating pseudo-random number easily [41]. However the collapsing of iterates of dynamical systems or at least the existence of very short periodic orbits, their non constant invariant measure, and the easily recognized shape of the function in the phase space should lead to avoid the use of such one-dimensional map (logistic, baker, or tent, etc.) or two dimensional map (Hénon, standard or Belykh, etc.) as a pseudo-random number generator (see [23] for a survey). However, the very simple implementation in computer program of chaotic dynamical systems led some authors to use it as a base of cryptosystem [2, 5]. They are topologically conjugate, that means they have similar topological properties (distribution, chaoticity, etc.) however due to the structure of number in computer realization their numerical behaviour differs drastically. Therefore the original idea here is to combine features of tent (T_μ) and logistic (L_μ) maps to achieve new map with improved properties, through combination in several topologies of network.

Looking to the equations we can inverse the shape of the graph of the tent map T on the step of logistic map L . Thus, our proposition has the form:

$$f_\mu(x) \equiv TL_\mu(x) = \mu|x| - \mu x^2 = \mu(|x| - x^2) \quad (6.13)$$

Recall that both logistic and tent maps are never used in cryptography because they have weak security (collapsing effect) [17, 47] if applied alone. Thus, systems are often used in modified form to construct PRNG [30, 45]. The Lozi system [36] provides method to increase randomness properties of the tent map over its coupling. In another way, we propose to couple T_μ map over combination with TL_μ map (6.13).

When used in more than one dimension, TL_μ map can be considered as a two variable map:

$$TL_\mu(x^{(1)}, x^{(2)}) = \mu(|x^{(1)}| - (x^{(2)})^2) \quad (6.14)$$

Hence it possible to define a mapping M_p from $[-1, 1]^p \rightarrow [-1, 1]^p$

$$M_p \begin{pmatrix} x_n^{(1)} \\ x_n^{(2)} \\ \vdots \\ x_n^{(p)} \end{pmatrix} = \begin{pmatrix} x_{n+1}^{(1)} \\ x_{n+1}^{(2)} \\ \vdots \\ x_{n+1}^{(p)} \end{pmatrix} = \begin{cases} T_\mu(x_n^{(1)}) + TL_\mu(x_n^{(1)}, x_n^{(2)}) \\ T_\mu(x_n^{(2)}) + TL_\mu(x_n^{(2)}, x_n^{(3)}) \\ \vdots \\ T_\mu(x_n^{(p)}) + TL_\mu(x_n^{(p)}, x_n^{(1)}) \end{cases} \quad (6.15)$$

Note that, the system dynamics is unstable and trajectories quickly spread out. Therefore, to solve the problem of holding dynamics in the bound $[-1, 1]^p$ the following injection mechanism has to be used:

$$\begin{array}{l} \text{if } x_{n+1}^{(i)} < -1 \\ \quad \text{then add 2} \\ \text{if } x_{n+1}^{(i)} > 1 \\ \quad \text{then subtract 2} \end{array} \quad (6.16)$$

in this case for $1 \leq i \leq p$, points come back from $[-3, 3]^p$ to $[-1, 1]^p$.

Used in conjunction with T_μ the TL_μ function allows to establish mutual influence between system states. The function is attractive because it performs contraction and stretching distance between states improving chaotic distribution. Thus, TL_μ function is a powerful tool to change dynamics.

The coupling of the simple states has excellent effect on chaos achieving, because:

- Simple states interact with global system dynamics, being a part of it.
- The states interaction has the global effect.

Hence, if we use TL_μ to make impact on dynamics of the simple maps then excellent effect on chaoticity and randomness could be achieved. The proposed function improve complexity of a simple map. The question is how to study the received system. Poincaré was one of the first who used graphical analysis of the complex systems. We will use also graphical approach to study new chaotic systems, but not only, other theoretical assessing functions are involved in our study.

Note that the system (6.15) can be seen in the scope of a general point of view, introducing constants k^i which generalize considered topologies. It is called alternative if $k^i = +1$, $1 \leq i \leq p$, or non-alternative if $k^i = +1$, $1 \leq i \leq p$; or $k^i = -1$, $1 \leq i \leq p$. It can be a mix of alternative and non-alternative if $k^i = +1$ or -1 randomly.

$$M_p \begin{pmatrix} x_n^{(1)} \\ x_n^{(2)} \\ \vdots \\ x_n^{(p)} \end{pmatrix} = \begin{pmatrix} x_{n+1}^{(1)} \\ x_{n+1}^{(2)} \\ \vdots \\ x_{n+1}^{(p)} \end{pmatrix} = \begin{cases} T_\mu(x_n^{(1)}) + k^1 \times TL_\mu(x_n^{(1)}, x_n^{(2)}) \\ T_\mu(x_n^{(2)}) + k^2 \times TL_\mu(x_n^{(2)}, x_n^{(3)}) \\ \vdots \\ T_\mu(x_n^{(p)}) + k^p \times TL_\mu(x_n^{(p)}, x_n^{(1)}) \end{cases} \quad (6.17)$$

In this paper we will discuss only systems exhibiting the best properties for CPRNG.

6.3.1 2-D Topologies

The initial purpose of new CPRNG design was to obtain excellent uniform distribution, successfully passing randomness and chaoticity tests. Thus we propose to consider firstly two 2-D models: alternative ($k^1 = -1, k^2 = 1$) and non-alternative ($k^1 = k^2 = 1$). However, coupling between states by TL_μ can be made in different ways:

1. Ring coupling with two choices:

$$TL_\mu^{RC}(x^{(1)}, x^{(2)}) = \begin{cases} T_\mu(x^{(1)}) - L_\mu(x^{(2)}) \\ T_\mu(x^{(2)}) - L_\mu(x^{(1)}) \end{cases} \quad (6.18)$$

or

$$TL_\mu^{RC}(x^{(2)}, x^{(1)}) = \begin{cases} T_\mu(x^{(2)}) - L_\mu(x^{(1)}) \\ T_\mu(x^{(1)}) - L_\mu(x^{(2)}) \end{cases} \quad (6.19)$$

2. Simple coupling with also two choices:

$$TL_\mu^{SC}(x^{(1)}, x^{(2)}) = \begin{cases} T_\mu(x^{(1)}) - L_\mu(x^{(2)}) \\ T_\mu(x^{(1)}) - L_\mu(x^{(2)}) \end{cases} \quad (6.20)$$

or

$$TL_\mu^{SC}(x^{(2)}, x^{(1)}) = \begin{cases} T_\mu(x^{(2)}) - L_\mu(x^{(1)}) \\ T_\mu(x^{(2)}) - L_\mu(x^{(1)}) \end{cases} \quad (6.21)$$

The general form of the new 2-D map we consider is as follow:

$$M_p \begin{pmatrix} x_n^{(1)} \\ x_n^{(2)} \end{pmatrix} = \begin{pmatrix} x_{n+1}^{(1)} \\ x_{n+1}^{(2)} \end{pmatrix} = \begin{cases} T_\mu(x_n^{(1)}) + k^1 \times TL_\mu(x^{(i)}, x^{(j)}) \\ T_\mu(x_n^{(2)}) + k^2 \times TL_\mu(x^{(i')}, x^{(j')}) \end{cases}$$

with $i, j, i', j' = 1$ or 2 and TL_μ being either TL_μ^{RC} or TL_μ^{SC} . Remark: Ring-coupling can be expected to higher dimensions but not the single case because we obtain the same expression of the function.

However, it is undesirable to use $TL_\mu^{SC}(x^{(1)}, x^{(2)})$ because (6.20) implies

$$\begin{aligned} M_p \begin{pmatrix} x_n^{(1)} \\ x_n^{(2)} \end{pmatrix} &= \begin{pmatrix} x_{n+1}^{(1)} \\ x_{n+1}^{(2)} \end{pmatrix} = \begin{cases} T_\mu(x_n^{(1)}) + k^1(T_\mu(x_n^{(1)}) - L_\mu(x_n^{(2)})) \\ T_\mu(x_n^{(2)}) + k^2(T_\mu(x_n^{(1)}) - L_\mu(x_n^{(2)})) \end{cases} \\ &\Leftrightarrow \begin{cases} x_{n+1}^{(1)} = k^1 L_\mu(x_n^{(2)}) \\ x_{n+1}^{(2)} = k^2 L_\mu(x_n^{(2)}) \end{cases} \end{aligned}$$

which is trivial.

If one uses $TL_\mu^{RC}(x^{(2)}, x^{(1)})$ alternative system then one of the states will have more “power” than another one, loosing good distribution of points property. For the same reason $TL_\mu^{SC}(x^{(1)}, x^{(2)})$ or $TL_\mu^{SC}(x^{(2)}, x^{(1)})$ non-alternative ($k = 1$) and $TL_\mu^{SC}(x^{(2)}, x^{(1)})$ alternative are not recommended to use.

Therefore, we will consider only two 2-D systems: $TTL_\mu^{RC}(x_n^{(2)}, x_n^{(1)})$ **non-alternative**:

$$TTL_\mu^{RC} : \begin{cases} x_{n+1}^{(1)} = 1 - \mu|x_n^{(1)}| + \mu(|x_n^{(2)}| - (x_n^{(1)})^2) \\ x_{n+1}^{(2)} = 1 - \mu|x_n^{(2)}| + \mu(|x_n^{(1)}| - (x_n^{(2)})^2) \end{cases} \quad (6.22)$$

and $TTL_\mu^{SC}(x_n^{(1)}, x_n^{(2)})$ **alternative**:

$$TTL_\mu^{SC} : \begin{cases} x_{n+1}^{(1)} = 1 - \mu|x_n^{(1)}| - \mu(|x_n^{(1)}| - (x_n^{(2)})^2) \\ x_{n+1}^{(2)} = 1 - \mu|x_n^{(2)}| + \mu(|x_n^{(1)}| - (x_n^{(2)})^2) \end{cases} \quad (6.23)$$

Both systems were selected because they have balanced contraction and stretching process between states allowing to achieve uniform distribution of the chaotic dynamic.

6.3.2 Randomness Study of the New Maps TTL_μ^{RC} and TTL_μ^{SC}

We are now assessing the randomness of both selected maps. The associated dynamical system is considered to be random and could be applied to cryptosystems if the chaotic generator meets the requirements 1–8 on Fig. 6.1 which are described in Sect. 1.3. If one of the criterion is not satisfied the behavior is less random than expected.

As it has been summarized in the scheme (Fig. 6.1) a generator could be taken into consideration for cryptography application if and only if every criterion is satisfied.

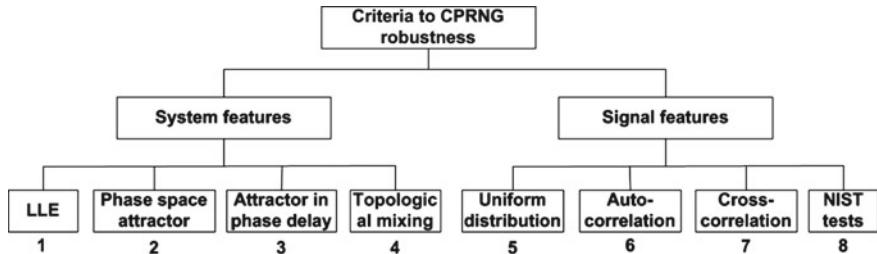


Fig. 6.1 The main criteria for PRNG robustness

Chaotic map behavior primarily depends on the initial guess x_0 and “control” parameter μ . However, the dependence versus the initial guess, x_0 has less importance when the global phase portrait is scrutinized. Thus, to study the dependency of parameter μ a bifurcation diagram is an appropriate tool. To create the diagram for the new map, a particular initial value of x_0 is randomly selected, and the map is iterated for a given μ . A certain number of firstly generated points is cut off to remove the transient part of the iterated points, and the following points are plotted. Afterwards, the process is repeated incrementing slightly μ .

To plot the bifurcation diagram for the 2-D systems TTL_{μ}^{RC} non-alternative (Fig. 6.2) and TTL_{μ}^{SC} alternative (Fig. 6.3), 10,000 iterations are generated for each initial value and the first 1000 points are cut off as transient. Thus, 9,000 points are plotted for each μ parameter. The graphs are the same for $x^{(1)}$ and $x^{(2)}$.

For both graphs starting from $\mu = 0$ to $\mu = 0.25$, we can observe a period 1 (i.e. a fixed point). Then the steady-state response undergoes a so-called pitchfork bifurcation to period 2. Following bifurcation undergoes multiple periods. At higher μ values, the behavior is generally chaotic. However, for TTL_{μ}^{RC} near $\mu = 1.1$ (Fig. 6.2) periodic windows appear. The subsequent intervals show perfect chaotic dynamics.

Fig. 6.2 Bifurcation diagram of 2-D new map:
 TTL_{μ}^{RC} non-alternative (6.22)

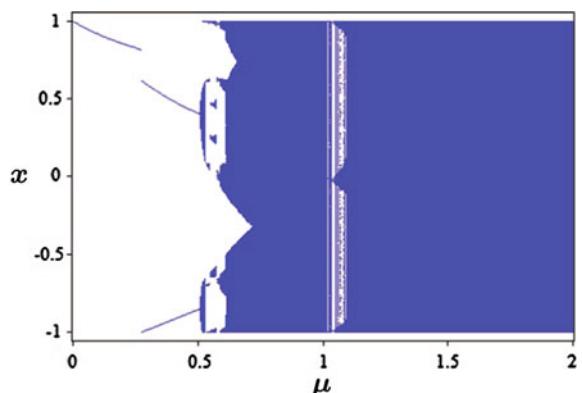
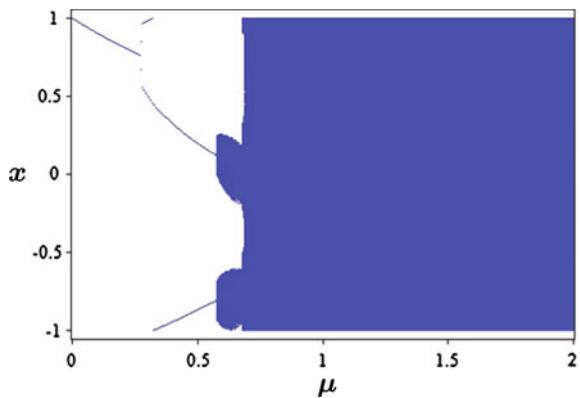


Fig. 6.3 Bifurcation diagram of 2-D new map: TTL_{μ}^{SC} alternative (6.23)



Bifurcation diagrams are very useful analysis tools for studying the behavior of nonlinear maps as well as control parameters impact on the dynamic. A complementary study of chaos is the graph of Lyapunov exponent.

The Lyapunov exponent is a measure of the system sensitivity to initial conditions. The function of Lyapunov exponent λ is the characteristic of chaotic behavior in nonlinear maps. If $\lambda > 0$ the system exhibits chaotic behaviour.

Let us observe the graphics of Lyapunov exponent for TTL_{μ}^{RC} non-alternative (Fig. 6.4) and TTL_{μ}^{SC} alternative (Fig. 6.5) maps. For the plotting 10,000 iterations were taken. The μ parameter is selected from 0.5 to 2. The list of points formed with μ is described on horizontal coordinate and the measure λ is on the vertical coordinate.

Graphs of the Lyapunov exponent are in exact agreement with bifurcations one. The measure λ is positive indicating chaotic dynamics which increases demonstrating the strongest chaos at $\mu = 2$.

The study demonstrates that TTL_{μ}^{RC} non-alternative (Fig. 6.4) and TTL_{μ}^{SC} alternative (Fig. 6.5) maps exhibit the best chaotic behavior characteristic when $\mu = 2$,

Fig. 6.4 Largest Lyapunov exponent for 2-D TTL_{μ}^{RC} non-alternative map (6.22)

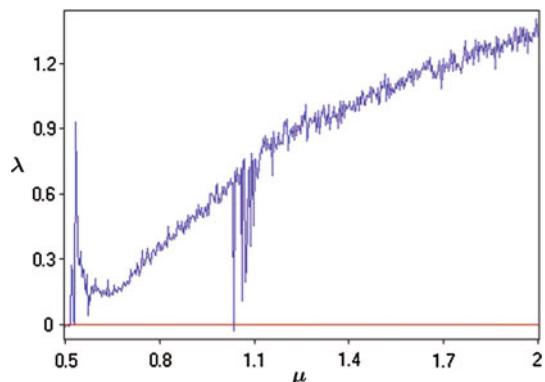
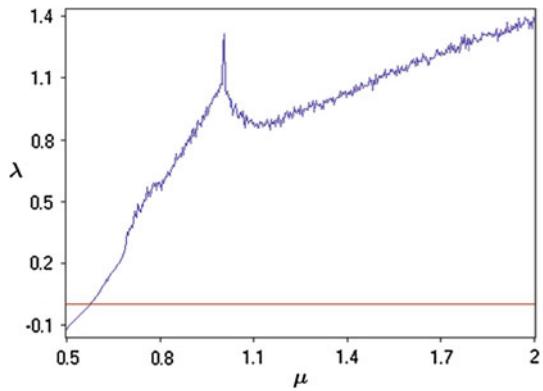


Fig. 6.5 Largest Lyapunov exponent for 2-D TTL_{μ}^{SC} alternative map (6.23)



therefore we will continue our study fixing the parameter to this value. On the graphs for any given initial point x_0 trajectories will look like chaotic. Hence, we can study an attractor in phase space and phase delay.

Let us plot the attractor in phase space [34]: $x_n^{(1)}$ versus $x_n^{(2)}$ to analyse the points distribution. Observing graphs of chaotic attractor we can make decision about complexity, notice weakness or infer the randomness nature [12]. To plot the attractor 3×10^4 points have been generated, 10^4 points of the transient regime have been cut off.

The graphs of the attractor in phase space for TTL_2^{RC} non-alternative (Fig. 6.6a) and TTL_2^{SC} alternative (Fig. 6.6b) maps are quite different. The first one has well scattered points on all the pattern, but there are some more “concentrated” regions forming curves on the graph. We will search answer to the questions: “Why there are more concentrated regions? From where curves creates?”, by considering the injection mechanism.

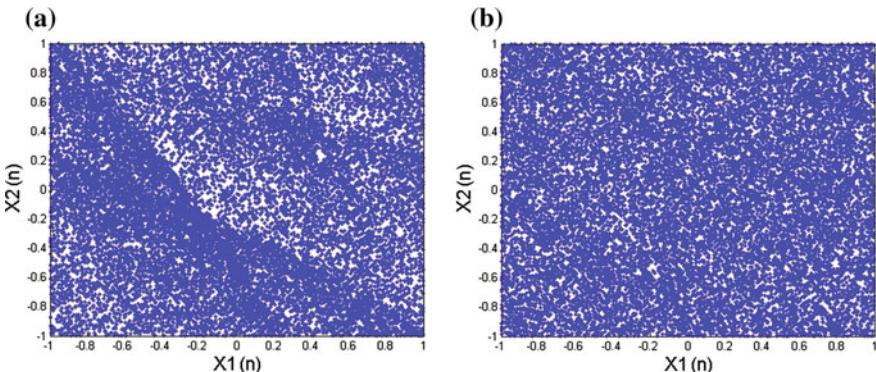


Fig. 6.6 Phase space attractor of 2-D new maps, 2×10^4 points are generated. **a** TTL_2^{RC} non-alternative (6.22). **b** TTL_2^{SC} alternative (6.23)

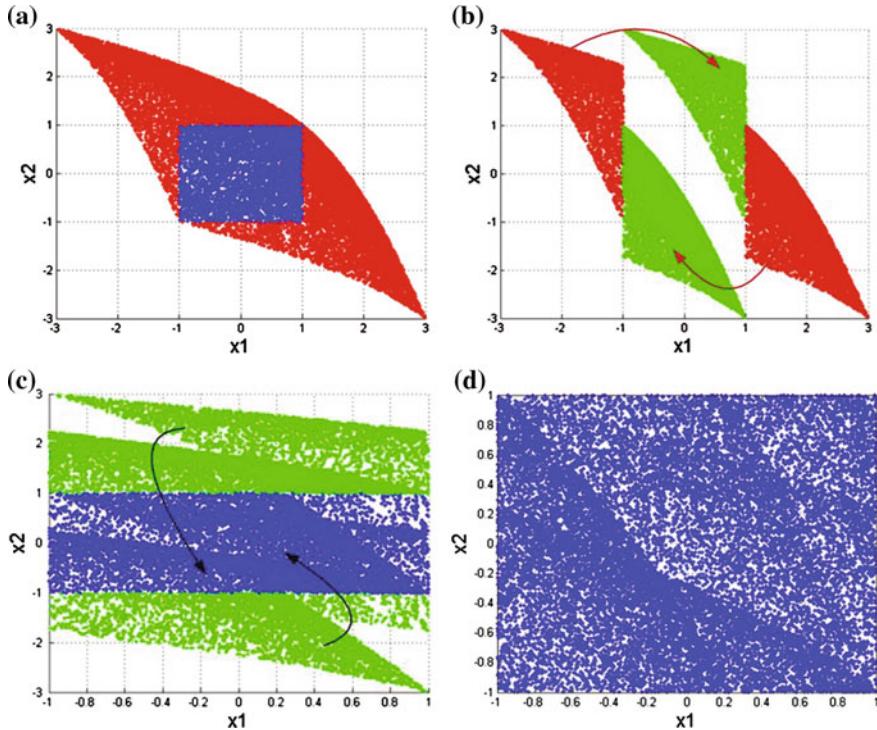


Fig. 6.7 Injection mechanism $[-3, 3]^2 \Rightarrow [-1, 1]^2$ for TTL_2^{RC} non-alternative map. **a** 2-D chaotic map without adding/subtraction, **b** injection $x_n^{(1)}$ to the torus $[-1, 1]^2$, **c** injection $x_n^{(2)}$ to the torus $[-1, 1]^2$, **d** results after passing injection mechanism

Without this mechanism dynamics goes out of the square $[-1, 1]^2$ (Fig. 6.7a). The maximal distance that points are reaching is 3 and the minimal is -3 . Thus, equations (6.16) are preserved, however their influence to the dynamics is different versus the Lozi system [21]. For the plotting, 2×10^4 points have been generated, 77 % of the points are scattered out of the $[-1, 1]^2$. The mechanism consists of p -steps for a p -dimensional system in each step the value 2 is added or subtracted to the variables if the dynamics goes out of the bounds (6.16). On the first step 69 % points are injected to the interval (Fig. 6.7b) after passing second injection step (Fig. 6.7c) all points are driven base to the square $[-1, 1]^2$ (Fig. 6.7d). Therefore mechanism adds non-linearity and complexity to the system which is an advantage from the security point of view, in the case of cryptographic use.

The graphs of the attractor in phase space for TTL_2^{SC} alternative map looks uniformly distributed on the plain pattern without any visible concentrated regions (Fig. 6.6b). The injection mechanism impact on the points distribution is given on the Fig. 6.8.

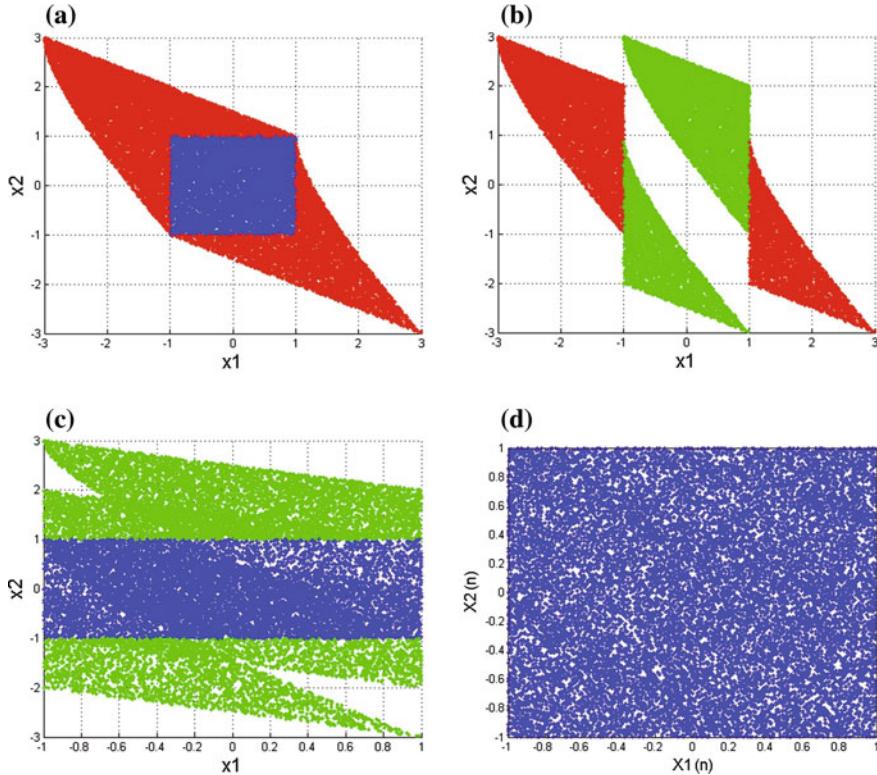


Fig. 6.8 Injection mechanism $[-3, 3]^2 \Rightarrow [-1, 1]^2$ for TTL_2^{SC} alternative map. **a** 2-D chaotic map without adding/subtraction, **b** injection $x_n^{(1)}$ to the torus $[-1, 1]^2$, **c** injection $x_n^{(2)}$ to the torus $[-1, 1]^2$, **d** results after passing injection mechanism

The quality of the entire cryptosystem mostly depends on PRNG and one of the most important things for robust PRNG is uniform distribution of generated values in the space (Criterion 5, Fig. 6.1). An approximated invariant measure gives the best picture of probability. Thus, the invariant measure (6.4) is used for precise study of the points distribution. Using the approximate density function the best picture of points density can be achieved. The graph of the function demonstrate distribution comparison between regions. The size of each of the boxes is measured by *step*. In other words the plain is divided $boxes[i, j]$ with square $step^2$ after the counts the number of points enter into the box $box[i, j]$ is counted.

For the approximation function the pattern was divided for 200 *boxes* or $step = 0.01, 10^9$ points were generated. Note that those values are the maximal possible used to calculate with a laptop computers. The graphs (Figs. 6.9 and 6.10) of the detail points distribution demonstrates that both systems don't excellent distribution in the phase space.

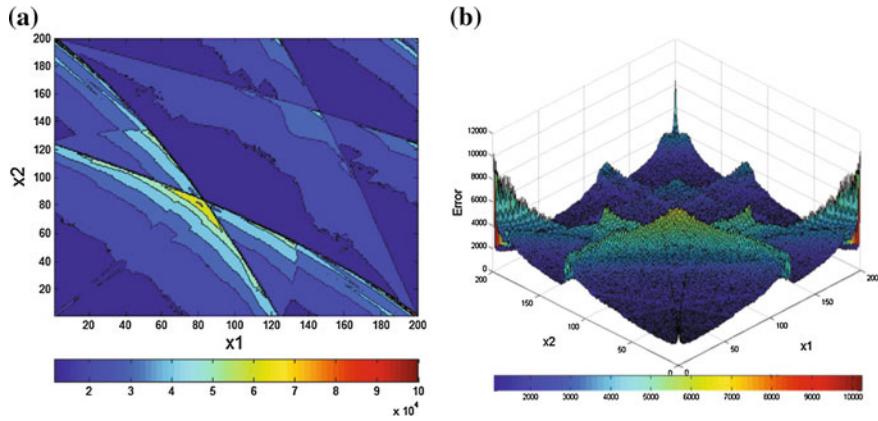


Fig. 6.9 Approximate density function of TTL_2^{RC} non-alternative map, where $step = 0.01$, 10^9 points are generated

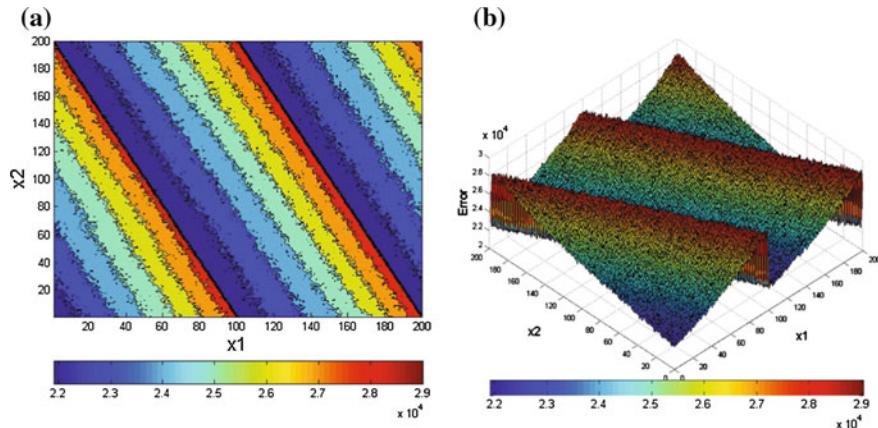


Fig. 6.10 Approximate density function of TTL_2^{SC} alternative map, where $step = 0.01$, 10^9 points are generated

It was noticed that some parts of the graph (Fig. 6.8b) are perfectly joined, giving us idea to improve points density using some correction in equations.

6.3.3 A New 2-D Chaotic PRNG

Considering the results of Sect. 1.3.2, it seems possible to improve the randomness of the 2-D topology. We observe that two regions (top-green and right-red) on the Fig. 6.8b could be pretty connected. First, let us rewrite the mapping TTL_μ^{SC} alternative (6.23) where $\mu = 2$ as follow:

$$TTL_2^{SC}(x_n^{(1)}, x_n^{(2)}) = \begin{cases} x_{n+1}^{(1)} = 1 + 2(x_n^{(2)})^2 - 4|x_n^{(1)}| \\ x_{n+1}^{(2)} = 1 - 2(x_n^{(2)})^2 + 2(|x_n^{(1)}| - |x_n^{(2)}|) \end{cases} \quad (6.24)$$

The first problem is that top green coloured region occurs after injection is applied. Thus, we develop the system (6.24) in such way that green coloured region “stays” in such position without injection mechanism. Secondly, we need to reduce the width of the region. Evidently, it is possible to achieve this need by reducing the impact of the state x^1 , with the new following map:

$$MTTL_2^{SC}(x_n^{(1)}, x_n^{(2)}) = \begin{cases} x_{n+1}^{(1)} = 1 + 2(x_n^{(2)})^2 - 2|x_n^{(1)}| \\ x_{n+1}^{(2)} = 1 - 2(x_n^{(2)})^2 + 2(|x_n^{(1)}| - |x_n^{(2)}|) \end{cases} \quad (6.25)$$

and the injection mechanism (6.16) is used as well, but restricted to 3 phases:

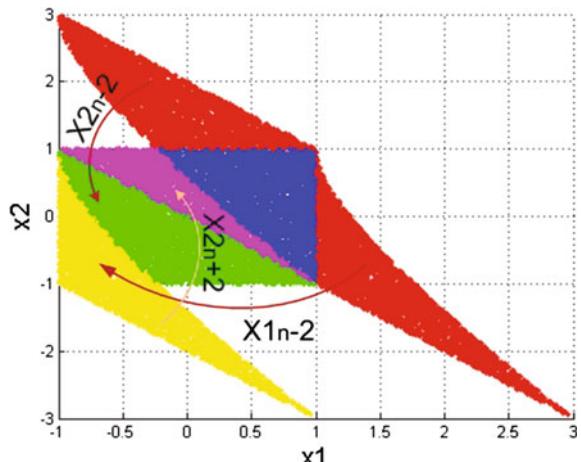
$$\begin{aligned} & \text{if } x_{n+1}^{(1)} > 1 \text{ then subtract 2} \\ & \text{if } x_{n+1}^{(2)} < -1 \text{ then add 2} \\ & \text{if } x_{n+1}^{(2)} > 1 \text{ then subtract 2} \end{aligned} \quad (6.26)$$

The results of the modifications are demonstrated on Figs. 6.11, 6.12 and 6.13. The injection mechanism in 3 phases (Fig. 6.11) pulled regions in an excellent way. The techniques used, greatly improve the points density in the phase space (Figs. 6.12, 6.13).

The numerical results of the errors distributions (Fig. 6.14) shows excellent distribution till 10^9 points which is limited by the classical computer power. Moreover, the largest Lyapunov exponent is equal to 0.5905 indicating strong chaotic behavior.

The graph (Fig. 6.14) shows straight error reducing that proves uniform points distribution.

Fig. 6.11 Injection mechanism (6.26) of $MTTL_2^{SC}$ alternative map



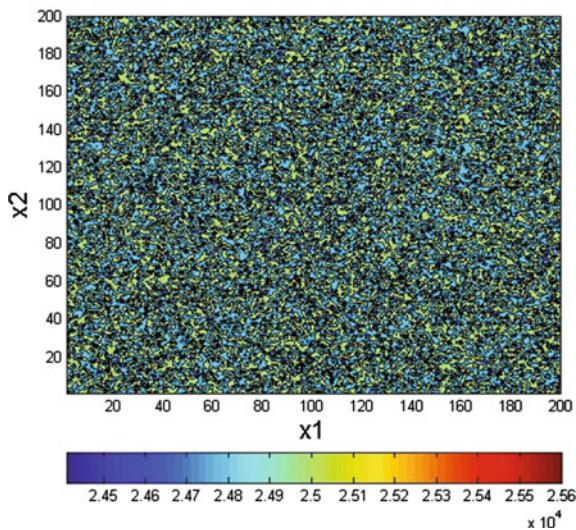


Fig. 6.12 Approximate density function of $MTTL_2^{SC}$ alternative map, where $step = 0.01$, 10^9 points are generated

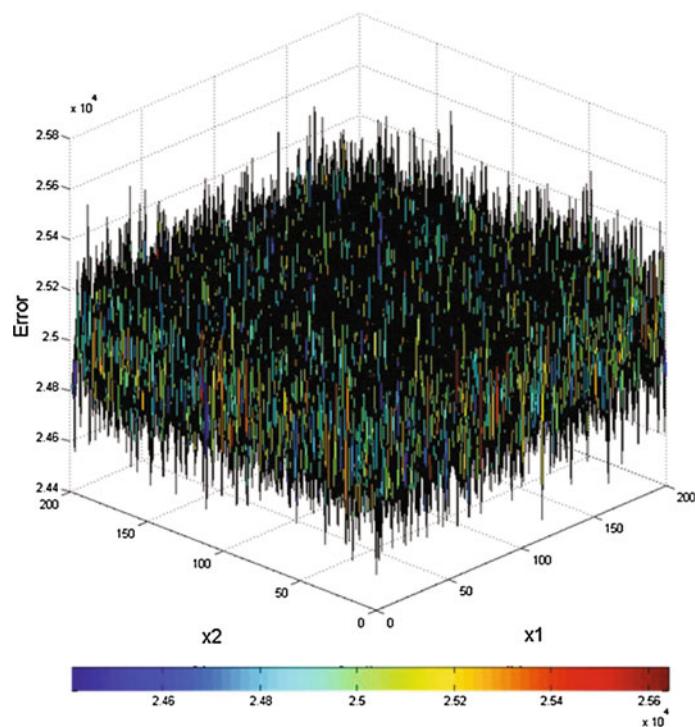


Fig. 6.13 Approximate density function in 3D of $MTTL_2^{SC}$ alternative map, where $step = 0.01$, 10^9 points are generated

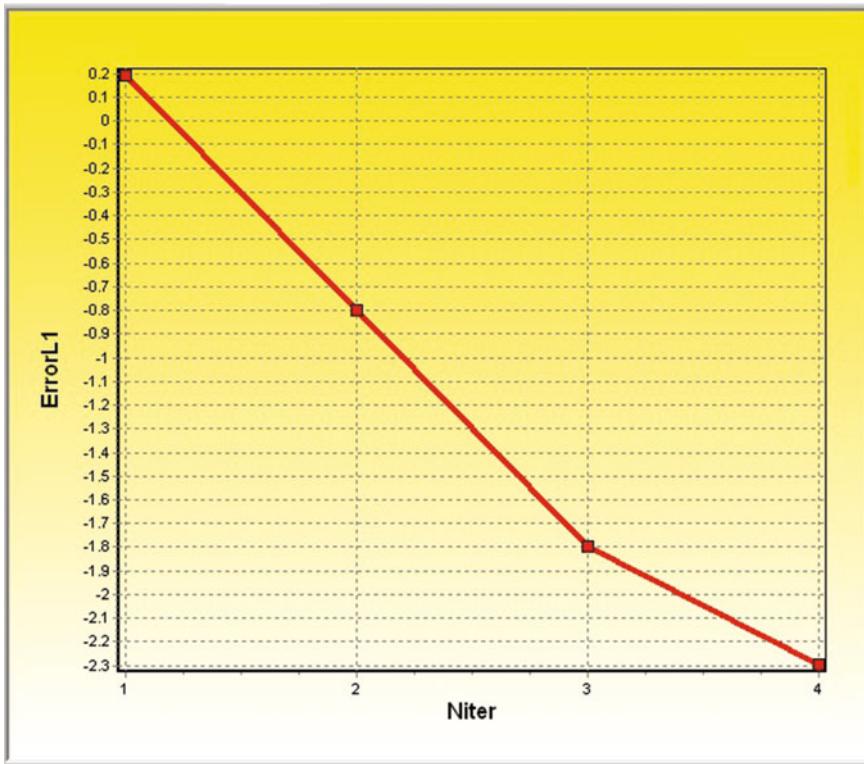


Fig. 6.14 Approximate distribution errors (6.6), for the system (6.25)

The points distribution of the attractor in phase delay is quite good as well (Figs. 6.15 and 6.16), where the plotting of 10^9 points are generated. On the Fig. 6.17b tent distribution is recognized for $x^{(2)}$ variable but for encryption we need only output of one state (in our case $x^{(1)}$). Both states make strong impact on itself and for the global dynamics reaching significant points distribution on the torus and chaoticity.

The $MTTL_2^{SC}$ alternative map is ring- and auto-coupled. Since one state takes part on creating dynamics of other one, both auto-correlation and cross-correlation have to be analysed for dependency and repeatability. The results of the 2-dimensional system are represented on the Figs. 6.18 and 6.19. The same excellent results are on the Fig. 6.18 for autocorrelation, and on the Fig. 6.19 for cross-correlation, where the sequences on the graphs are near zero. In addition, approximate distribution errors (6.6), (6.7) and (6.8) numerically indicate homogeneous density (Table 6.1).

Topologically mixing means the system capability to progress over a short period of time. The system from any given initial region or open set of its phase space will ultimately mixed up with any other region so that it is impossible to predict system evolution.

Fig. 6.15 Attractor in the phase delay $(x_n^{(1)}, x_{n+1}^{(1)})$, 10^9 points are generated, for the system (6.25)

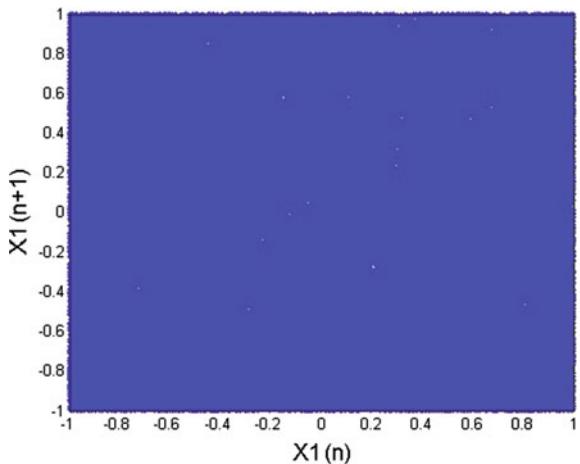
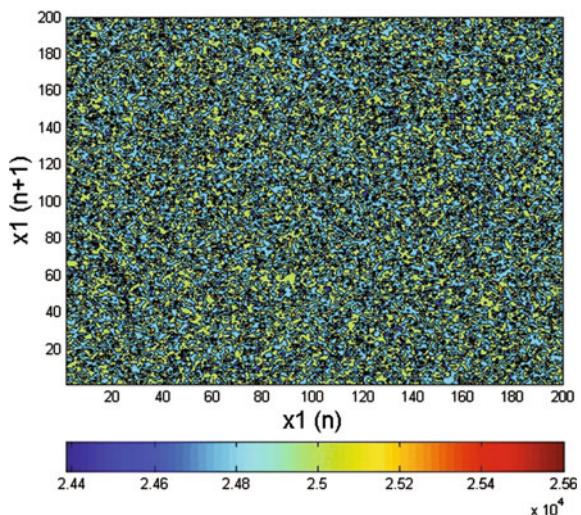


Fig. 6.16 Attractor in the phase delay $(x_n^{(1)}, x_{n+1}^{(1)})$, box-method, 10^9 points are generated, for the system (6.25)



Here we represent graphical analysis of the 2-D $MTTL_2^{SC}$ alternative map for topological mixing. The square $[0, 1]^2$ is divided into 4 quadrants and each of them are split in boxes as well ($A2, B2, C2, \dots, O2$). 5×10^3 points have been generated in each of the boxes (Fig. 6.20) and on the Fig. 6.21a–e it is showed where the points from the initial boxes ($A1, B1, C1, \dots, O1$) of quadrant are mapped.

From the Fig. 6.21 it can be seen that points are distributed everywhere over the torus $[-1, 1]^2$, and it is hard to predict the next point or to find the previous one. The system is perfectly mixing because the regions are superimposed to each other. For example if we take some point of the $A2$ box (Fig. 6.20) the next point will fall down to the $A2$ region (Fig. 6.21a). The blue coloured region on the Fig. 6.21a passes through the boxes: $O1, I1, P1, C1, B1, E1, H1, M4, N4$ (Fig. 6.22), that means the

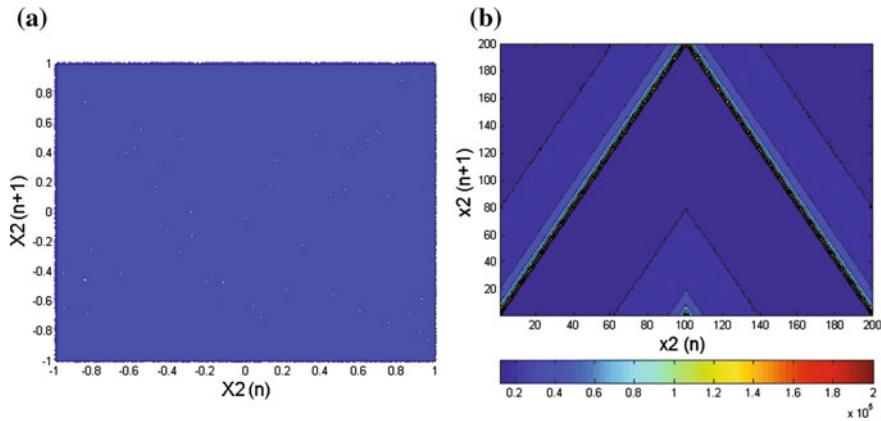


Fig. 6.17 Attractor in the phase delay, 10^9 points are generated. **a** $(x_n^{(2)}, x_{n+1}^{(2)})$. **b** Box-method

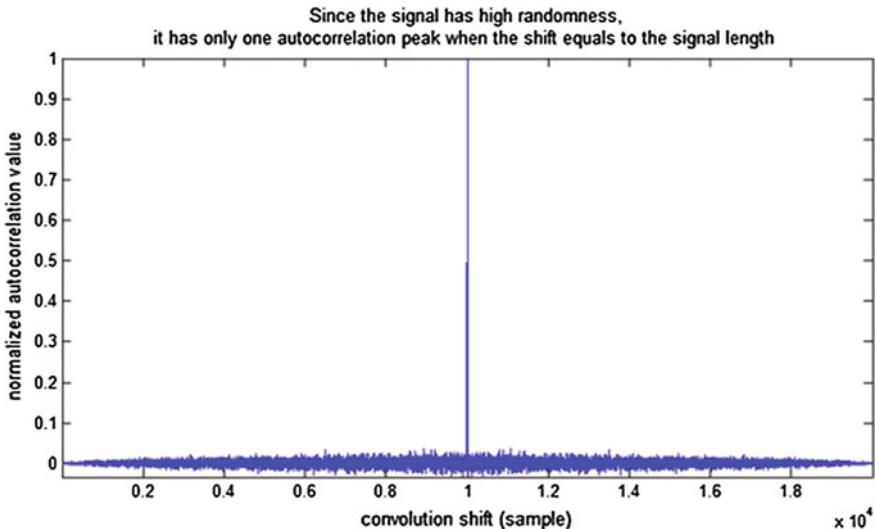


Fig. 6.18 State auto-correlation analysis of the $MTTL_2^{SC}$ alternative map

next points will fall down somewhere on the regions corresponding to these boxes (Fig. 6.21a–e). With all next iterations, they mix more complexly; the behavior becomes unpredictable and eventually looks like scattered points everywhere across the space. Colours and letters overlapping on the graphs vividly demonstrate that arbitrarily close points in some periods of time will have vastly different behaviors which means mixing. This phenomenon is quantified through the value of Largest Lyapunov exponent. The arbitrarily taken points which are far alone will ultimately approach looking nearly the same only for several iterations means mixing as well. Since the new map implies of strong chaos, the phase space is thoroughly mixed

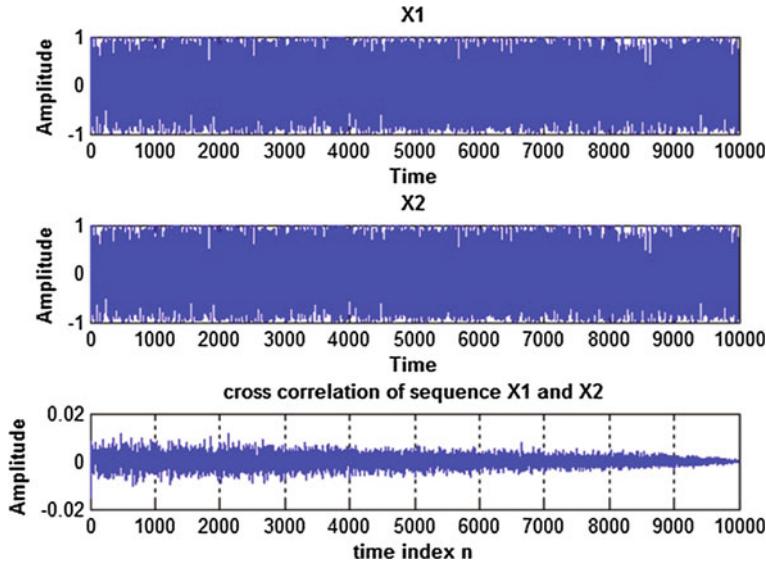


Fig. 6.19 Correlation between states of the $MTTL_2^{SC}$ alternative map

Table 6.1 Approximate distribution errors (6.6, 6.7 and 6.8), for the system (6.25) in phase space

Points	$x^{(i)}x^{(j)}$	ErrorL1	ErrorL2	ErrorL3
10^4	$x^{(1)}x^{(2)}$	1.55830000000011	3.9967999999983	16
10^6	$x^{(1)}x^{(2)}$	0.15812000000055	0.395695199999969	1.56
10^8	$x^{(1)}x^{(2)}$	0.015989099999995	0.0401757055999971	0.1748
10^9	$x^{(1)}x^{(2)}$	0.0050540619999996	0.0040140246800009	0.04916

together after a quite short time. In a forthcoming paper we will quantify this mixing, building a corresponding Markov transition matrix as in [25].

NIST tests are used to verify randomness and system capability to resist main attacks. As it was earlier discussed the advantage of the binary sequences has to be approximately the same as of the truly random number generator. NIST tests are more fully cover the statistical tests. Long time the tests are used to prove PRNG robustness. NIST tests require only binary sequences, thus 4×10^6 points were generated, the first 5×10^5 were cut off. The rest of the sequence was converted to binary form according to the standard IEEE-754 (32 bit single precision floats).

Both states of the generator successfully passed NIST tests demonstrating strong randomness being robustness against numerous statistical attacks (Fig. 6.23). Moreover, we can say that generated sequences look like truly random. Thus, if the adversary looks at the sequence it will be difficult to distinguish it from a truly random generator.

6.4 A New Higher-Dimensional Map

Higher dimensional systems allow to achieve the best randomness, chaoticity and points distribution, because there are more perturbations and nonlinear mixing in it. Usually, 3 or more dimensions are enough to create robust random sequences. Thus, it is an advantage if the system could increase its dimensions. Since, $MTTL_2^{SC}$ alternative map cannot be in higher dimension, we describe how to improve randomness, best points distribution and more complex dynamics than $TTL_2^{RC}(x^{(2)}, x^{(1)})$ alternative map (6.22).

The best way to achieve randomness from chaos is to couple states with auto and ring-coupling [22]. After applying the conditions the higher dimension map takes form as follow:

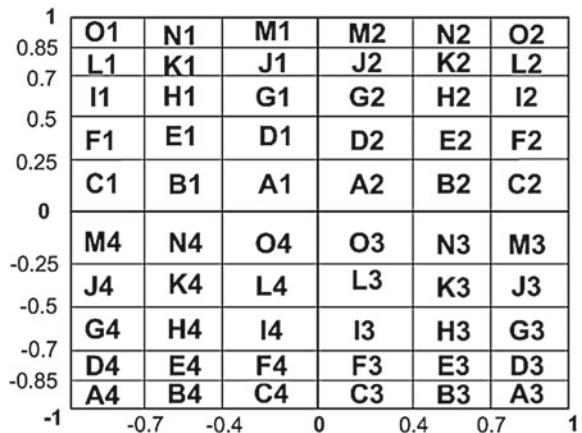
$$TTL_2^{RC} : \begin{cases} x_{n+1}^{(1)} = 1 - 2|x_n^{(1)}| + 2(|x_n^{(2)}| - (x_n^{(1)})^2) \\ x_{n+1}^{(2)} = 1 - 2|x_n^{(2)}| + 2(|x_n^{(3)}| - (x_n^{(2)})^2) \\ \vdots \\ x_{n+1}^{(p)} = 1 - 2|x_n^{(p)}| + 2(|x_n^{(1)}| - (x_n^{(p)})^2) \end{cases} \quad (6.27)$$

The injection is applied as well by verifying each of the state for diverging, in the case if, the injection is used.

Note, each of the states has to satisfy requirements and chaoticity. Therefore, the 3-D and 4-D system were studied for criteria 1–8 (Fig. 6.1) independently for the each states and in correlation between them. All of the tests have been successfully passed with improving results whereas dimension is higher. Here we demonstrate only more significant and important tests.

First of all, the points distribution is the best tool to demonstrate the system evolution with increasing dimension. Therefore, to draw the plot 10^9 points were

Fig. 6.20 Initial boxes (A, B, C, \dots, O) in the four quadrants



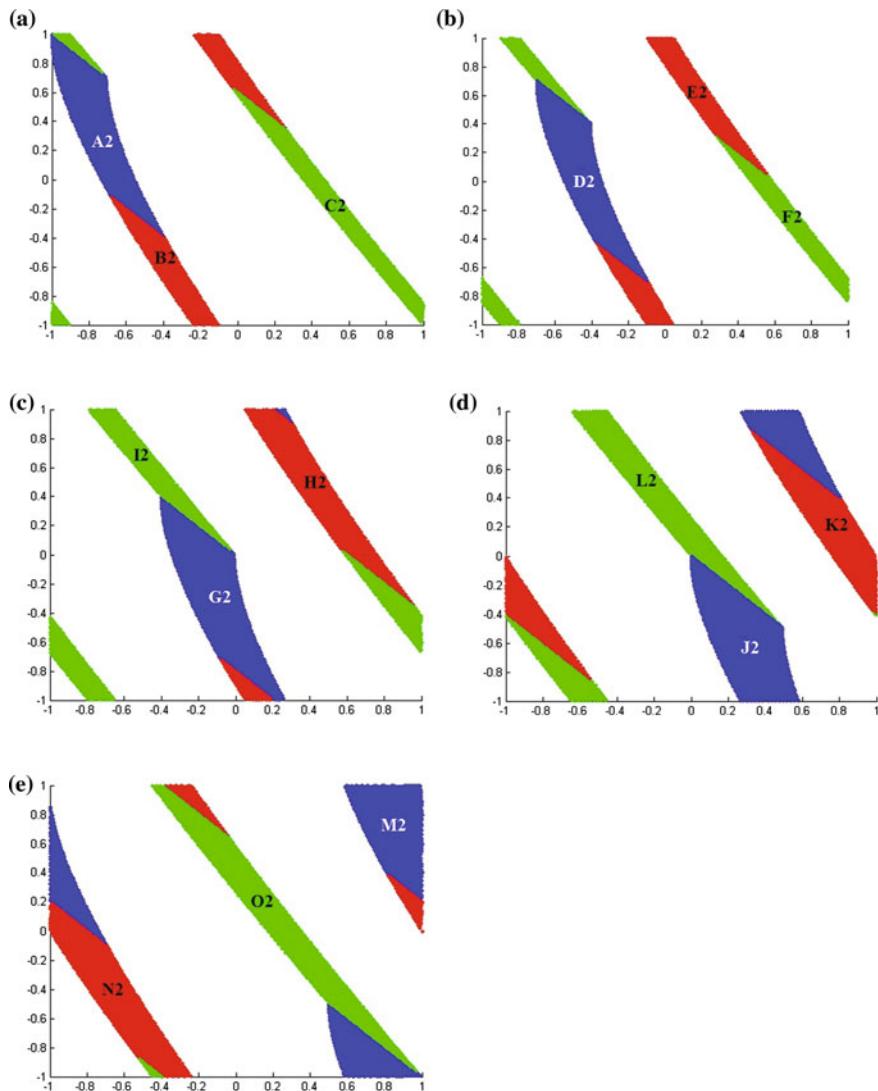


Fig. 6.21 Topological mixing

generated for: 2-D, 3-D and 4-D system. The invariant measure was calculated with distribution error results fixing on the iterations: 10^4 , 10^6 , 10^8 and 10^9 . The graph (Fig. 6.24) shows improving the points distribution the space.

After generating 10^6 points for 2-D system sequences become repeatable because the errors no longer decrease. This phenomenon may be due to long periodic orbits attracting the behaviour of iterated points. For 3-D system period is longer but is locked after 10^9 generated points because errors should be reduced 10 times on each

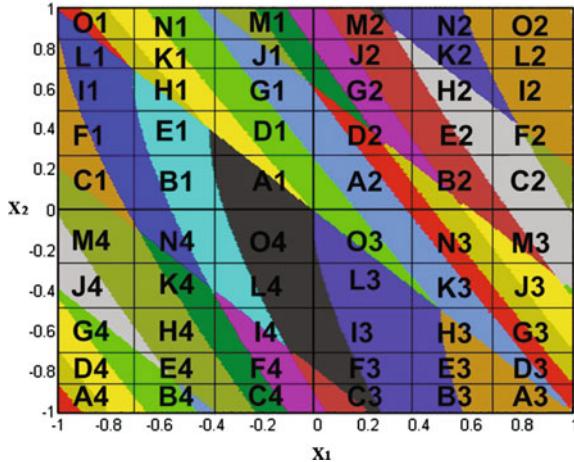


Fig. 6.22 Mixing boxes ($A \dots O$) and regions (coloured) in the phase space $(x_n^{(1)}, x_n^{(2)})$

(a)

RESULTS FOR THE UNIFORMITY OF P-VALUES AND THE PROPORTION OF PASSING SEQUENCES												
generator is <data/Modified TL_{\mu}^{\{SC\}} alternative map_x1.txt>												
C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	P-VALUE	PROPORTION	STATISTICAL TEST
8	8	11	9	10	8	11	15	11	9	0.897763	100/100	Frequency
13	13	12	7	11	10	12	9	5	8	0.678686	99/100	BlockFrequency
6	7	5	12	16	12	12	9	14	7	0.191687	100/100	CumulativeSums
8	10	12	6	14	12	9	6	12	11	0.678686	100/100	Runs
14	11	12	10	15	5	6	13	8	6	0.236810	99/100	LongestRun
9	6	13	10	7	10	11	11	12	11	0.897763	97/100	Rank
11	12	6	19	4	11	11	13	8	5	0.037566	97/100	FFT
7	9	13	14	12	9	9	11	7	9	0.816537	100/100	NonOverlappingTemplate
10	11	15	10	11	9	12	6	11	5	0.595549	98/100	OverlappingTemplate
11	10	5	7	5	13	16	5	13	15	0.059884	100/100	Universal
14	6	11	10	7	9	13	12	8	10	0.739918	98/100	ApproximateEntropy
2	9	7	8	5	7	5	5	8	7	0.689019	63/63	RandomExcursions
5	8	4	4	6	4	4	11	6	11	0.222869	63/63	RandomExcursionsVariant
12	10	12	13	7	8	7	7	6	18	0.171867	99/100	Serial
9	13	11	12	7	9	7	16	7	9	0.534146	99/100	LinearComplexity

(b)

RESULTS FOR THE UNIFORMITY OF P-VALUES AND THE PROPORTION OF PASSING SEQUENCES												
generator is <data/Modified TL_{\mu}^{\{SC\}} alternative map_x2.txt>												
C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	P-VALUE	PROPORTION	STATISTICAL TEST
18	6	8	12	9	6	7	10	11	13	0.191687	98/100	Frequency
12	7	12	7	3	11	13	10	13	12	0.366918	98/100	BlockFrequency
15	14	8	6	8	13	7	10	9	10	0.494392	98/100	CumulativeSums
12	15	11	8	7	12	9	5	8	13	0.474986	98/100	Runs
9	12	13	13	9	14	9	6	8	7	0.637119	100/100	LongestRun
8	12	8	10	13	15	10	6	7	11	0.616305	98/100	Rank
8	12	9	15	9	8	17	9	9	4	0.181557	99/100	FFT
7	12	7	12	6	9	15	12	7	13	0.437274	100/100	NonOverlappingTemplate
9	12	11	3	16	8	10	13	10	8	0.289667	99/100	OverlappingTemplate
9	13	10	6	8	8	11	10	11	14	0.816537	99/100	Universal
7	24	9	7	7	8	8	17	7	6	0.000347	98/100	ApproximateEntropy
2	4	2	5	5	7	2	13	4	8	0.011791	52/52	RandomExcursions
5	4	8	5	2	1	8	6	4	9	0.191687	52/52	RandomExcursionsVariant
6	10	8	7	15	15	15	8	8	8	0.236810	100/100	Serial
7	9	11	11	6	15	7	11	8	15	0.419021	99/100	LinearComplexity

Fig. 6.23 MTTL₂^{SC} alternative map successfully passed NIST tests. **a** $x^{(1)}$. **b** $x^{(2)}$

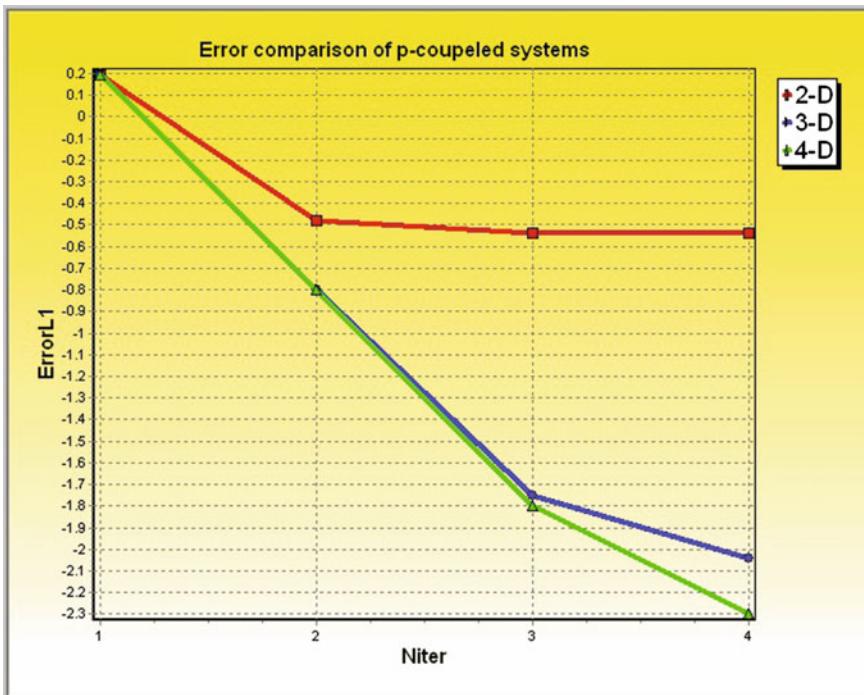


Fig. 6.24 Density error (6.6) for: 2-D, 3-D and 4-D $TTL_2^{RC}(x^{(2)}, x^{(1)})$ alternative map

Table 6.2 Numerical results of the error points distribution for 3-D $TTL_2^{RC}(x^{(2)}, x^{(1)})$ alternative map

Points	$x^{(i)}x^{(j)}$	ErrorL1	ErrorL2	ErrorL3
10^4	$x^{(1)}x^{(2)}$	1.55695000000012	3.9871999999827	16
10^4	$x^{(1)}x^{(3)}$	1.55960000000011	4.0287999999834	16
10^4	$x^{(2)}x^{(3)}$	1.55850000000012	4.011199999983	16
10^6	$x^{(1)}x^{(2)}$	0.160244000000057	0.40613359999969	1.56
10^6	$x^{(1)}x^{(3)}$	0.159324000000056	0.40040639999964	1.72
10^6	$x^{(2)}x^{(3)}$	0.159722000000056	0.40181279999966	1.64
10^8	$x^{(1)}x^{(2)}$	0.017516779999997	0.0483318551999966	0.1788
10^8	$x^{(1)}x^{(3)}$	0.017657899999997	0.0488421623999967	0.1784
10^8	$x^{(2)}x^{(3)}$	0.017617139999997	0.0485752623999967	0.1836
10^9	$x^{(1)}x^{(2)}$	0.0090892079999996	0.0125199035839995	0.0772
10^9	$x^{(1)}x^{(3)}$	0.0090351620000002	0.0124306507039994	0.08368
10^9	$x^{(2)}x^{(3)}$	0.0090724099999998	0.0124629701279995	0.07804

$100 \times length$. Note, when length goes to infinity ($length = 10^{11}$, for example) the error no longer decreases. The systems distribution errors comparison is demonstrated on Fig. 6.24.

The robust PRNG implies the points to have equal chance to be chosen. Thus, the system appears to be unpredictable. The precise comparison can be made by numerical calculation to compare deviation from etalon distribution: ErrorL1 (6.6), ErrorL2 (6.7) and ErrorL3 (6.8). The Table 6.2 displays numerical results for 3-D map and in (Table 6.3) for 4-D $TTL_2^{RC}(x^{(2)}, x^{(1)})$ alternative map are demonstrated.

The numerical results demonstrate harmony of the points density between states. Moreover, the NIST tests prove it randomness (Fig. 6.25).

Table 6.3 Numerical results of the error points distribution for 4-D $TTL_2^{RC}(x^{(2)}, x^{(1)})$ alternative map

Points	$x^{(i)}x^{(j)}$	ErrorL1	ErrorL2	ErrorL3
10^4	$x^{(1)}x^{(2)}$	1.55720000000011	3.999199999983	16
10^4	$x^{(1)}x^{(3)}$	1.55655000000012	3.9687999999831	16
10^4	$x^{(1)}x^{(4)}$	1.55495000000012	3.9551999999832	20
10^4	$x^{(2)}x^{(3)}$	1.5581000000001	4.006399999983	16
10^4	$x^{(2)}x^{(4)}$	1.5576000000001	4.004799999983	16
10^4	$x^{(3)}x^{(4)}$	1.55395000000012	3.9351999999834	16
10^6	$x^{(1)}x^{(2)}$	0.15857000000055	0.39843279999969	1.64
10^6	$x^{(1)}x^{(3)}$	0.159702000000056	0.40437759999966	1.68
10^6	$x^{(1)}x^{(4)}$	0.160002000000056	0.40510719999971	1.64
10^6	$x^{(2)}x^{(3)}$	0.158936000000056	0.39959359999971	1.52
10^6	$x^{(2)}x^{(4)}$	0.159348000000055	0.40184799999965	1.68
10^6	$x^{(3)}x^{(4)}$	0.158972000000057	0.39914879999965	1.72
10^8	$x^{(1)}x^{(2)}$	0.015983139999994	0.0400194487999969	0.1608
10^8	$x^{(1)}x^{(3)}$	0.016025539999995	0.04038192319997	0.1772
10^8	$x^{(1)}x^{(4)}$	0.016036659999995	0.0404230903999969	0.1852
10^8	$x^{(2)}x^{(3)}$	0.016044199999995	0.0403678407999969	0.1732
10^8	$x^{(2)}x^{(4)}$	0.015879279999996	0.0396031839999973	0.1612
10^8	$x^{(3)}x^{(4)}$	0.015810119999993	0.03918319999997	0.164
10^9	$x^{(1)}x^{(2)}$	0.0050723279999997	0.00404898352000012	0.0524
10^9	$x^{(1)}x^{(3)}$	0.0051505899999998	0.00415637283200005	0.05388
10^9	$x^{(1)}x^{(4)}$	0.0050473119999992	0.00399370235200004	0.05932
10^9	$x^{(2)}x^{(3)}$	0.0050579599999996	0.00400627627200004	0.05516
10^9	$x^{(2)}x^{(4)}$	0.0051483659999991	0.00416637750400014	0.05228
10^9	$x^{(3)}x^{(4)}$	0.0050373479999993	0.00397888753600011	0.05112

(a)

C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	P-VALUE	PROPORTION	STATISTICAL TEST
8	14	8	9	10	9	11	12	6	13	0.779188	100/100	Frequency
11	9	9	8	6	15	7	13	9	13	0.574903	100/100	BlockFrequency
14	6	13	7	11	5	10	11	9	14	0.401199	100/100	CumulativeSums
12	10	7	7	16	8	13	7	13	7	0.366918	99/100	CumulativeSums
16	9	7	11	14	12	6	13	7	5	0.181557	100/100	Runs
13	9	14	11	11	8	9	12	5	8	0.678686	100/100	LongestRun
14	9	7	8	9	16	9	12	6	10	0.455937	100/100	Rank
13	4	9	11	7	4	10	12	19	11	0.037566	100/100	FFT
14	8	8	9	8	15	10	11	8	8	0.699313	100/100	NonOverlappingTemplate
14	15	12	10	6	9	13	7	3	11	0.162606	99/100	OverlappingTemplate
8	7	11	16	9	12	10	9	7	11	0.678686	100/100	Universal
13	11	10	12	6	12	12	14	6	4	0.304126	97/100	ApproximateEntropy
5	5	6	9	2	7	5	8	9	6	0.637119	62/62	RandomExcursions
6	2	4	9	6	11	6	5	6	7	0.407091	62/62	RandomExcursionsvariant
13	8	15	8	12	9	7	15	8	5	0.275709	99/100	Serial
13	6	15	12	11	6	15	8	8	6	0.213309	99/100	Serial
9	6	8	13	8	11	10	11	12	12	0.883171	99/100	LinearComplexity

(b)

C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	P-VALUE	PROPORTION	STATISTICAL TEST
7	5	12	14	10	9	12	16	8	7	0.289667	99/100	Frequency
7	7	9	10	6	10	14	8	10	19	0.137282	99/100	BlockFrequency
8	2	9	16	13	9	13	9	7	14	0.090936	99/100	CumulativeSums
5	8	14	11	11	11	14	5	10	11	0.437274	99/100	CumulativeSums
6	16	13	11	9	10	8	7	11	9	0.554420	100/100	Runs
9	13	6	9	14	10	8	11	12	8	0.779188	99/100	LongestRun
9	8	14	6	12	12	8	10	8	13	0.719747	100/100	Rank
10	10	17	5	9	13	14	10	6	6	0.153763	99/100	FFT
9	7	9	13	9	10	10	14	6	13	0.719747	100/100	NonoverlappingTemplate
5	9	12	7	7	12	12	13	12	11	0.637119	99/100	OverlappingTemplate
12	16	8	7	9	10	7	12	8	11	0.616305	99/100	Universal
8	16	6	12	11	13	5	7	13	9	0.249284	99/100	ApproximateEntropy
4	8	4	6	8	5	7	8	9	7	0.804337	66/66	RandomExcursions
4	7	7	8	2	8	6	8	7	9	0.602458	66/66	RandomExcursionsvariant
11	10	10	18	6	5	11	12	10	7	0.213309	100/100	Serial
8	11	10	10	12	11	10	9	9	10	0.998821	98/100	Serial
10	7	13	11	8	7	11	14	11	8	0.798139	99/100	LinearComplexity

Fig. 6.25 NIST tests for, **a** 3-D $TTL_2^{RC}(x^{(2)}, x^{(1)})$ alternative map. **b** 4-D $TTL_2^{RC}(x^{(2)}, x^{(1)})$ alternative map

6.5 Conclusion

In this paper we have proposed the original idea to couple two well-known chaotic maps (tent and logistic one), which considered separately—don't exhibit the required features for encryption purposes. However, the new coupling changed qualitatively the overall system behavior, because the maps used with injection mechanism and coupling between states increase their complexity.

We have explored several topologies and finally proposed a new 2-D CPRNG. The proposed model with injection mechanism allows to puzzle perfectly the pieces of the chaotic attractor, like a true random generator. To achieve the best distribution in the phase space, the modified form $MTTL_2^{SC}$ alternative map has been proposed. The new map exhibits excellent features due to the injection mechanism and enables the uniform density in the state space. The system exhibits strong nonlinear dynamics, demonstrating great sensitivity to initial conditions. It generates an infinite range of intensive chaotic behavior with large positive Lyapunov exponent values. Moreover, $MTTL_2^{SC}$ successfully passed all required tests: cross-correlation, autocorrelation,

LLE, NIST tests, uniform attractor on the phase space and phase delay. The system analysis and the dynamics evolution by bifurcation diagram and topological mixing proved the complex behavior. The system orbits exhibited complex behavior with perfect mixing. The study demonstrated totally unpredictable dynamics making the system strong-potential candidate for high-security applications. Finally, the dimension of the TTL_{μ}^{RC} non-alternative map is easily increased whenever it is necessary to reach the strongest security requirements as shown in Sect. 1.4.

References

1. Alvarez, G., Li, S.: Some basic cryptographic requirements for chaos-based cryptosystems. *Int. J. Bifurc. Chaos* **16**, 2129–2151 (2006)
2. Ariffin, M.R.K., Noorani, M.S.M.: Modified Baptista type chaotic cryptosystem via matrix secret key. *Phys. Lett. A* **372**, 5427–5430 (2008)
3. Banerjee, S., Kastha, D., Das, S., Vivek, G., Grebogi, C.: Robust chaos—the theoretical formulation and experimental evidence. In: Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS '99), vol. 5, pp. 293–296 (1999)
4. Banks, J., Brooks, J., Cairns, G., Davis, G., Stacey, P.: On Devaney's definition of chaos. *Am. Math. Mon.* **99**, 332–334 (1992)
5. Baptista, M.S.: Cryptography with chaos. *Phys. Lett. A* **240**, 50–54 (1998)
6. Dachselt, F., Schwarz, W.S.: Chaos and cryptography. *IEEE Trans. Circuits Syst. I: Fundam. Theory Appl.* **48**, 1498–1509 (2001)
7. Dogan, R., Murgan, A.T., Ortmann, S., Glesner, M.: Searching for robust chaos in discrete time neural networks using weight space exploration. *IEEE Int. Conf. Neural Netw.* **2**, 688–693 (1996)
8. Dowell, E.H., Pezeshki, C.: On the understanding of chaos in Duffings equation including a comparison with experiment. *J. Appl. Mech.* **55**, 5–9 (1986)
9. Feigenbaum, M.J.: The universal metric properties of nonlinear transformations. *J. Stat. Phys.* **21**(6), 669–706 (1979)
10. Feki, M.: An adaptive chaos synchronization scheme applied to secure communication. *Chaos Solitons Fractals* **18**(1), 141–148 (2003)
11. Frey, D.R., Schwarz, W.: Chaotic digital encoding: an approach to secure communication. *IEEE Trans. Circuits Syst. II: Analog Digit. Signal Process.* **40**, 660–666 (1993)
12. Garasym, O., Taralova, I., Lozi, R.: Application of nonlinear dynamics to chaotic PRNG design. In: 2014 International Conference on European Conference Iteration Theory (ECIT), vol.20 (2014)
13. Heidari-Bateni, G., McGillem, C.D.: A chaotic direct-sequence spread-spectrum communication system. *IEEE Trans. Commun.* **42**, 1524–1527 (1994)
14. Holenstein, T.: Pseudorandom generators from one-way functions: a simple construction for any hardness. In: *Theory Cryptography*, pp. 443–461 (2009)
15. Hong, Z., Ling, X.: Generating chaotic secure sequences with desired statistical properties and high security. *Int. J. Bifurc. Chaos* **7**, 205–213 (1997)
16. Katz, O., Ramon, D.A., Wagner, I.A.: A robust random number generator based on a differential current-mode chaos. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **16**, 1677–1686 (2008)
17. Lanford III, O.E.: Informal remarks on the orbit structure of discrete approximations to chaotic maps. *Exp. Math.* **7**, 317–324 (1998)
18. Li, C., Chen, G.: Chaos in the fractional order Chen system and its control. *Chaos Solitons Fractals* **22**, 549–554 (2004)
19. Li, C.Y., Chen, Y.H., Chang, T.Y., Deng, L.Y., Kiwing, T.: Period extension and randomness enhancement using high-throughput reseeding-mixing PRNG. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **20**(2), 385–389 (2012)

20. Liebert, W., Schuster, H.G.: Proper choice of the time delay for the analysis of chaotic time series. *Phys. Lett. A* **142**, 107–111 (1989)
21. Lozi, R.: Chaotic pseudo random number generators via ultra weak coupling of chaotic maps and double threshold sampling sequences, In: ICCSA 2009, 3rd Conference on Complex Systems and Applications, pp. 20–24 (2009)
22. Lozi, R.: Emergence of randomness from chaos. *Int. J. Bifurc. Chaos* **22**(02), 1250021 (2012)
23. Lozi, R.: Can we trust in numerical computations of chaotic solutions of dynamical systems? *Topol. Dyn. Chaos, World Sci. Ser. Nonlinear Sci. Ser. A* **84**, 63–98 (2013)
24. Lozi, R., Cherrier, E.: Noise-resisting ciphering based on a chaotic multi-stream pseudo-random number generator, In: 2011 International Conference for Internet Technology and Secured Transactions (ICITST), pp. 91–96 (2011)
25. Lozi, R., Taralova, I.: From chaos to randomness via geometric undersampling. *ESAIM: Proc Surv.* **46**, 177–195 (2014)
26. Ma, H.G., Han, C.Z.: Selection of embedding dimension and delay time in phase space reconstruction. *Front. Electr. Electron. Eng. China* **1**(1), 111–114 (2006)
27. May, R.: Stability and Complexity of Models Ecosystems. Princeton University Press, Princeton (1973)
28. May, R.: Biological populations with overlapping generations: stable points, stable cycles, and chaos. *Science* **186**(4164), 645–647 (1974)
29. Menezes, A.J., Van Oorschot, P.C.: Handbook of applied cryptography. CRC Press, Boca Raton (1996)
30. Nejati, H., Beirami, A., Massoud, Y.: A realizable modified tent map for true random number generation. In: Circuits Systems, MWSCAS, vol. 10, pp. 621–624 (2008)
31. Nillsen, R.: Randomness and recurrence in dynamical systems. *AMC* **10**, 12–30 (2010)
32. Noura, H., El Assad, S., Vladeanu, C.: Design of a fast and robust chaos-based cryptosystem for image encryption. In: 2010 8th International Conference on Communications (COMM), pp. 423–426 (2010)
33. Odibat, Z.M., Corson, N., Aziz-Alaoui, M.A., Bertelle, C.: Synchronization of chaotic fractional-order systems via linear control. *Int. J. Bifurc. Chaos* **20**, 81–97 (2010)
34. Pichler, L., Pradlwarter, H.J.: Evolution of probability densities in the phase space for reliability analysis of non-linear structures. *Struct. Saf.* **31**, 316–324 (2009)
35. Reingold, O.: Theory of cryptography. In: 6th Theory of Cryptography Conference, TCC, 15–17 March (2009)
36. Rojas, A., Taralova, I., Lozi, R.: New alternate ring-coupled map for multirandom number generation. *J. Nonlinear Syst. Appl.* **4**(1), 64–69 (2013)
37. Rukhin, A., Soto, J., Nechvatal, J., Smid, M., Barker, E.: A statistical test suite for random and pseudorandom number generators for cryptographic applications. Booz-Allen and Hamilton Inc Mclean Va (2010)
38. Sato, S., Sano, M., Sawada, Y.: Practical methods of measuring the generalized dimension and the largest Lyapunov exponent in high dimensional chaotic systems. *Prog. Theor. Phys.* **77**, 1–5 (1987)
39. Singh, A., Gilhotra, R.: Data security using private key encryption system based on arithmetic coding. *Int. J. Netw. Secur. Appl. (IJNSA)* **3**, 58–67 (2011)
40. Sprott, J.C.: Chaos and Time-Series Analysis, p. 69. Oxford University Press, Oxford (2003)
41. Sudret, B.: Global sensitivity analysis using polynomial chaos expansions. *Reliab. Eng. Syst. Saf.* **93**, 964–979 (2008)
42. Sundarapandian, V., Pehlivan, I.: Analysis, control, synchronization, and circuit design of a novel chaotic system. *Math. Comput. Model.* **12**, 1904–1915 (2012)
43. Thiffeault, J.L., Finn, M.D., Gouillart, E., Hall, T.: Topology of chaotic mixing patterns. *Chaos Interdiscip. J. Nonlinear Sci.* **18**, 033123 (2008)
44. Wang, S., Kuang, J., Li, J., Luo, Y., Lu, H., Hu, G.: Chaos-based secure communications in a large community. *Phys. Rev. E* **66**, 065202 (2002)
45. Wong, W.K., Lee, L.P., Wong, K.W.: A modified chaotic cryptographic method. In: Communications and Multimedia Security Issues of the New Century, pp. 123–126 (2001)

46. Xiong, J., Yang, Z.: Chaos caused by a topologically mixing map. *Int. Cent. Theory Phys.* (1991)
47. Yuan, G., Yorke, J.A.: Collapsing of chaos in one dimensional maps. *Phys. D: Nonlinear Phenom.* **136**, 18–30 (2000)
48. Zaher, A.A., Abdulnasser, A.R.: On the design of chaos-based secure communication systems. *Commun. Nonlinear Sci. Numer. Simul.* **16**(9), 3721–3737 (2011)
49. Zhou, X., Tang, X.: Research and implementation of RSA algorithm for encryption and decryption. In: 2011 6th International Forum on Strategic Technology (IFOST), vol. 1, pp. 1118–1121 (2011)

Chapter 7

Distributed Finite-Time Cooperative Control of Multi-agent Systems

Yu Zhao, Guanghui Wen and Guanrong Chen

Abstract The distributed finite-time consensus problems for second-order multi-agent systems are studied in Sect. 7.1. Then, in Sect. 7.2, the distributed finite-time containment protocols are designed for second-order multi-agent systems with multiple nonlinear dynamic leaders. Finally, for multiple Euler-Lagrange systems, some finite-time tracking protocols are proposed in Sect. 7.3.

During the past two decades, distributed cooperative control of autonomous agents has become a promising research direction and received increasing attentions from different fields. Research on this topic aims to understand how various group behaviors can emerge as a result of local interactions among individuals. Distributed cooperative control has wide applications in different engineering areas, such as design of sensor networks, distributed tracking of multiple vehicles, containment control, formation flying of spacecraft, and cooperative surveillance [1–3], due to such advantages it offers as low cost, high robustness and easy maintenance [4].

In the context of distributed coordinating control, a group of autonomous agents, by coordination with each other via communication or sensing networks, can perform certain challenging tasks which cannot be well accomplished by a single agent. As one of the important and fundamental research issues for multi-agent systems, consensus problem has been extensively studied over the past few years. The objective is to develop some distributed control policies using only local relative information to ensure that the states of the agents reach an agreement on certain quantities of interest. Two pioneering works on consensus are [5, 6]. A theoretical explanation was

Y. Zhao (✉)

School of Automation, Northwestern Polytechnical University, Xi'an 710072,
People's Republic of China
e-mail: yuzhao5977@gmail.com

G. Wen

Department of Mathematics, Southeast University, Nanjing 210096,
People's Republic of China
e-mail: wenguanghui@gmail.com

G. Chen

Department of Electronic Engineering, City University of Hong Kong,
Hong Kong SAR, China
e-mail: eegchen@cityu.edu.hk

provided in [5] for the alignment behavior observed in the Vicsek model [7], while in [6] a general framework of the consensus problem for networks of integrators was proposed. Since then, a variety of consensus algorithms have been suggested to solve consensus problems in different scenarios, see [1–3, 8–14] and references therein. According to the number of leaders in a multi-agent network, existing consensus problems can be roughly categorized into three classes, namely, consensus without a leader (i.e., leaderless consensus), consensus with a single leader, and consensus in the presence of multiple leaders. Consensus with a single leader is also called distributed tracking or leader-following consensus. The objective of consensus tracking is to design some distributed protocols such that the states of the followers equipped with these protocols can track those of the leader. Consensus tracking problems for multi-agent systems with integrator-type dynamics were studied in [1, 15, 16], where some effective algorithms were proposed for the followers to track a dynamic leader. Further, [3, 17] extended the simple linear dynamics of agents into general linear systems. In the presence of multiple leaders, the containment control problem arises, where the objective is to drive the states of all followers to move into a convex hull spanned by those of the multiple leaders. Containment control problems have also been extensively investigated for multi-agent systems with integrator-type dynamics [2, 18–20]. Specifically, in [18, 19], the followers are assumed to have double-integrator dynamics. Moreover, containment control problems for multiple Euler-Lagrange systems and nonlinear systems were studied in [21, 22], respectively.

Convergence rate, as a significant performance index for evaluating the effectiveness of the designed consensus algorithms, is a focal research topic in the area of consensus for multi-agent systems. Many researchers have endeavored into improving the convergence rate by enlarging the coupling strengths, optimizing the system gains, or designing better communication topologies [23–26]. However, the above-mentioned methods may only guarantee asymptotic consensus. In practical applications, it is often desirable to achieve consensus in finite time. Thus, it is essential to investigate finite-time consensus problems. Finite-time consensus problem was first studied in [27], where the agents are assumed to have first-order dynamics. Then, finite-time consensus problems with double-integrator dynamics and nonlinear dynamics were investigated in [28, 29], respectively. Furthermore, distributed finite-time tracking control and containment control were investigated in [30–32]. It is worth noting that most of the above-mentioned works focused on distributed finite-time consensus under the assumption that all agents have the same dynamics or the leaders' accelerations are zero. Generally, this assumption is too strict to be satisfied by practical multi-agent systems. Thus, it is important to solve the distributed finite-time tracking problem for multi-agent systems under leaders with non-zero accelerations.

Motivated by the above observations, this chapter investigates the distributed finite-time cooperative control problem for second-order multi-agent systems. The main contribution is that some distributed finite-time tracking protocols based on relative measurements are designed such that all followers with double-integrator dynamics can track those of a leader with double-integrator dynamics or nonlin-

ear dynamics in finite time. Furthermore, finite-time containment problem is also investigated.

7.1 Distributed Finite-Time Consensus of Second-Order Multi-agent Systems

In this section, distributed finite-time consensus and formation control problems for second-order multi-agent systems are addressed. First, a finite-time tracking protocol is designed by using both relative position and velocity information. Then, a new class of observer-based control algorithms are designed for achieving finite-time consensus tracking in the considered multi-agent systems with a single active leader, where each agent may only share its position states with the neighbors. Within the same context, the present control algorithms are extended to solve the finite-time formation control problem. At last, the finite-time leaderless consensus problem with saturated protocol is discussed.

7.1.1 Designing Protocols Based on both Relative Position and Velocity Measurements

In this subsection, distributed finite-time consensus tracking control for multi-agent systems is considered. For the convenience of analysis, the communication topology is assumed to satisfy the following two assumptions:

Assumption 7.1 The communication topology between the followers is undirected.

Assumption 7.2 For each follower, there exists at least one directed path from the leader to it.

Now consider a group of second-order integrator agents described by

$$\dot{q}_i = p_i, \quad \dot{p}_i = u_i, \quad i = 1, \dots, N, \quad (7.1)$$

where $q_i \in \mathbf{R}^n$ is the position of agent i , and $p_i \in \mathbf{R}^n$ is its velocity. The dynamics of the leader, labeled as $N + 1$, is given by

$$\dot{q}_{N+1}^d = p_{N+1}^d, \quad \dot{p}_{N+1}^d = 0, \quad (7.2)$$

with an initial position $q_{N+1}^d(0) = r$ and an initial velocity $p_{N+1}^d(0) = v$, which generates the desired target trajectory. It is assumed that the leader can only be observed by a small fraction of the followers. And, if follower i can obtain information from the leader, node i with $a_{i(N+1)} = 1$ is referred to as a pinned node. Denote the

Laplacian matrix between the followers as \mathbf{L}_p and $P = \text{diag}\{a_{1(N+1)}, \dots, a_{N(N+1)}\}$. The following property can be verified by using matrix theory.

Lemma 7.1 [33] Under Assumptions 7.1 and 7.2, matrix P has at least one positive entry and

$$\mathbf{L}_f = \mathbf{L}_p + P \quad (7.3)$$

is a positive definite symmetric matrix.

Definition 7.1 The finite-time consensus tracking problem is solved if and only if there exists $T > 0$, such that, $\lim_{t \rightarrow T} (q_i - q_{N+1}^d) = 0$, $\lim_{t \rightarrow T} (p_i - p_{N+1}^d) = 0$, and $q_i = q_{N+1}^d$, $p_i = p_{N+1}^d$, if $t > T$, for all $i = 1, \dots, N$.

Some basic results of finite-time control theory are first reviewed. Given a vector $x = (x_1, \dots, x_n)^T$, define

$$\begin{aligned} |x|^\alpha &= (|x_1|^\alpha, \dots, |x_n|^\alpha)^T, \\ \text{sig}(x)^\alpha &= (\text{sgn}(x_1)|x_1|^\alpha, \dots, \text{sgn}(x_n)|x_n|^\alpha)^T, \end{aligned} \quad (7.4)$$

where $\alpha \in \mathbf{R}$.

Lemma 7.2 [34] Consider the following dynamical system:

$$\begin{aligned} \dot{x}_1 &= x_2, \\ \dot{x}_2 &= -M(\text{sig}(x_1)^{\alpha_1} + \text{sig}(x_2)^{\alpha_2}), \end{aligned} \quad (7.5)$$

where $x_1, x_2 \in \mathbf{R}^n$, and M is a positive definite matrix. The zero equilibrium of system (7.5) is globally finite-time stable if $0 < \alpha_1 < 1$ and $\alpha_2 = \frac{2\alpha_1}{1+\alpha_1}$.

Lemma 7.3 [34] Consider the following dynamical system:

$$\begin{aligned} \dot{x}_1 &= x_2, \\ \dot{x}_2 &= -M(\text{sig}(x_1 + e_1)^{\alpha_1} + \text{sig}(x_2 + e_2)^{\alpha_2}), \\ \dot{e}_1 &= e_2 - k_1 \text{sig}(e_1)^{\sigma_1}, \\ \dot{e}_2 &= -k_2 \text{sig}(e_1)^{\sigma_1}, \end{aligned} \quad (7.6)$$

where $x_1, x_2, e_1, e_2 \in \mathbf{R}^n$, and M is a positive definite matrix. The zero equilibrium of system (7.6) is globally finite-time stable if $k_1 > 0, k_2 > 0, \frac{1}{2} < \sigma_1 < 1, \alpha_1 = 2\sigma_1 - 1, \alpha_2 = \frac{2\sigma_1 - 1}{\sigma_1}$, and $\sigma_2 = 2\sigma_1 - 1$.

Now, we investigate the finite-time consensus tracking problem with a leader using state feedback, leaving the study by using output feedback to the next subsection.

Consider a multi-agent system described by (7.1) and (7.2). It is first assumed in this subsection that the relative states between each agent and its neighbors can be used. The protocol based on relative state information is proposed as follows:

$$\begin{aligned} u_i = & -\text{sig}\left(\sum_{j=1}^N a_{ij}(q_i - q_j) + a_{i(N+1)}(q_i - q_{N+1}^d)\right)^{\alpha_1} \\ & -\text{sig}\left(\sum_{j=1}^N a_{ij}(p_i - p_j) + a_{i(N+1)}(p_i - p_{N+1}^d)\right)^{\alpha_2}, \quad i = 1, \dots, N, \end{aligned} \quad (7.7)$$

where $\alpha_1, \alpha_2 \in \mathbf{R}$ are parameters to be determined.

Theorem 7.1 Suppose that Assumptions 7.1 and 7.2 hold. Then, the positions and velocities of the followers under protocol (7.7) will converge to those of the leader in finite time, respectively, if

$$0 < \alpha_1 < 1, \quad \alpha_2 = \frac{2\alpha_1}{1 + \alpha_1}. \quad (7.8)$$

Proof By substituting (7.7) into (7.1), the closed-loop system of the multi-agent system is transformed into

$$\begin{aligned} \dot{q}_i &= p_i, \\ \dot{p}_i &= -\text{sig}\left(\sum_{j=1}^N a_{ij}(q_i - q_j) + a_{i(N+1)}(q_i - q_{N+1}^d)\right)^{\alpha_1} \\ &\quad -\text{sig}\left(\sum_{j=1}^N a_{ij}(p_i - p_j) + a_{i(N+1)}(p_i - p_{N+1}^d)\right)^{\alpha_2}, \end{aligned} \quad (7.9)$$

where $i = 1, \dots, N$. Denote $q = (q_1^T, \dots, q_N^T)^T$, $p = (p_1^T, \dots, p_N^T)^T$, $q^l = ((q_{N+1}^d)^T, \dots, (q_{N+1}^d)^T)^T$ and $p^l = ((p_{N+1}^d)^T, \dots, (p_{N+1}^d)^T)^T$. Then, it follows from (7.9) that the closed-loop system of the multi-agent system can be rewritten in a matrix form as

$$\begin{aligned} \dot{q} &= p, \\ \dot{p} &= -\text{sig}[(\mathbf{L}_p \otimes I)q + (P \otimes I)q^l]^{\alpha_1} \\ &\quad -\text{sig}[(\mathbf{L}_p \otimes I)p + (P \otimes I)p^l]^{\alpha_2}, \end{aligned}$$

where \mathbf{L}_p and P are defined in (7.3). Let $x = q - q^l$ and $y = p - p^l$, respectively. From (7.2), one has

$$\begin{aligned} \dot{x} &= y, \\ \dot{y} &= -\text{sig}[(\mathbf{L}_p \otimes I)(q - q^l) + (P \otimes I)(q - q^l)]^{\alpha_1} \\ &\quad -\text{sig}[(\mathbf{L}_p \otimes I)(p - p^l) + (P \otimes I)(p - p^l)]^{\alpha_2} \\ &= -\text{sig}[(\mathbf{L}_f \otimes I)x]^{\alpha_1} - \text{sig}[(\mathbf{L}_f \otimes I)y]^{\alpha_2}. \end{aligned}$$

Taking linear transformation $X = (\mathbf{L}_f \otimes I)x$ and $Y = (\mathbf{L}_f \otimes I)y$, one gets

$$\begin{aligned}\dot{X} &= Y, \\ \dot{Y} &= -(\mathbf{L}_f \otimes I)(\text{sig}(X)^{\alpha_1} + \text{sig}(Y)^{\alpha_2}).\end{aligned}\quad (7.10)$$

According to Lemma 7.2 and (7.8), system (7.10) is globally finite-time stable. Therefore, there exists $T > 0$, such that $\lim_{t \rightarrow T} X = \lim_{t \rightarrow T} Y = 0$. By Lemma 7.1, $(\mathbf{L}_f \otimes I)$ is positive definite. Thus, $\lim_{t \rightarrow T} x = \lim_{t \rightarrow T} y = 0$ since $X = (\mathbf{L}_f \otimes I)x$ and $Y = (\mathbf{L}_f \otimes I)y$, namely, q and p will converge to q^l and p^l in finite time, respectively. It implies that the global finite-time consensus tracking control problem of the multi-agent system (7.1) and (7.2) is solved by the finite-time protocol (7.7). This completes the proof.

Remark 7.1 A finite-time protocol based on relative full-state information is proposed, which assumes that the relative position and velocity information are available for protocol design. However, this assumption is strict in many applications where only the position but not the velocity can be measured. Therefore, a novel observer-based protocol is proposed in the following based only on relative position measurements.

7.1.2 Designing Observer-Based Protocols Without Velocity Measurement

In this subsection, finite-time consensus tracking control for multi-agent systems with a single leader is studied by using only relative output information. In what follows, the relative output measurements of other agents with respect to agent i are synthesized into a single signal as follows:

$$z_i = \sum_{j=1}^{N+1} a_{ij}(q_i - q_j), \quad i = 1, \dots, N, \quad (7.11)$$

where $a_{ij} = 1$ if agent i can obtain information from agent j but $a_{ij} = 0$ otherwise. Based on the above analysis, an observer-based consensus protocol is proposed as

$$\begin{aligned}\dot{\delta}_i &= \eta_i - k_1 \text{sig}(\delta_i - z_i)^{\sigma_1}, \\ \dot{\eta}_i &= \sum_{j \in \mathcal{N}_i} a_{ij}(u_i - u_j) - k_2 \text{sig}(\delta_i - z_i)^{\sigma_2}, \\ u_i &= -\text{sig}(\delta_i)^{\alpha_1} - \text{sig}(\eta_i)^{\alpha_2}, \quad i = 1, \dots, N,\end{aligned}\quad (7.12)$$

where k_1, k_2 are feedback gains and $\sigma_1, \sigma_2, \alpha_1, \alpha_2 \in \mathbf{R}$ are parameters to be determined. The term $\sum_{j \in \mathcal{N}_i} a_{ij}(u_i - u_j)$ in (7.12) denotes the information exchanges

between the protocol of agent i and those of its neighbors. Since the leader acts as a command generator, it is reasonable to assume that the control law u_{N+1} of the leader is zero, i.e., $u_{N+1} = 0$. Note that control protocol (7.12) is distributed, since it is based only on the relative information of neighboring agents.

Theorem 7.2 Suppose that Assumptions 7.1 and 7.2 hold. Then, the finite-time consensus tracking problem for system (7.1) with an active leader (7.2) is solved by protocol (7.12), if

$$\begin{aligned} k_1 &> 0, \quad k_2 > 0, \\ \frac{1}{2} < \sigma_1 < 1, \quad \sigma_2 &= 2\sigma_1 - 1, \quad \alpha_1 = 2\sigma_1 - 1, \quad \alpha_2 = \frac{2\sigma_1 - 1}{\sigma_1}. \end{aligned} \quad (7.13)$$

Proof Let $x_i = q_i - q^d$ and $y_i = p_i - p^d$, for all $i = 1, \dots, N$. According to (7.1) and (7.2), one gets

$$\begin{aligned} \dot{x}_i &= y_i, \\ \dot{y}_i &= u_i, \quad i = 1, \dots, N. \end{aligned} \quad (7.14)$$

Substituting (7.12) into (7.14) yields

$$\begin{aligned} \dot{x}_i &= y_i, \\ \dot{y}_i &= -\text{sig}(\delta_i)^{\alpha_1} - \text{sig}(\eta_i)^{\alpha_2}, \\ \dot{\delta}_i &= \eta_i - k_1 \text{sig}(\delta_i - z_i)^{\sigma_1}, \\ \dot{\eta}_i &= -\sum_{j=1}^N l_{ij} \text{sig}(\delta_j)^{\alpha_1} - \sum_{j=1}^N l_{ij} \text{sig}(\eta_j)^{\alpha_2} - k_2 \text{sig}(\delta_i - z_i)^{\sigma_2}, \end{aligned} \quad (7.15)$$

where $i = 1, \dots, N$.

Let $q = (q_1^T, \dots, q_N^T)^T$, $p = (p_1^T, \dots, p_N^T)^T$, $z = (z_1^T, \dots, z_N^T)^T$ and $\underline{q}^d = (\mathbf{1} \otimes I)q_{N+1}^d$. Then, according to (7.11), one has

$$\begin{aligned} z &= (\mathbf{L}_p \otimes I)q + (P \otimes I)(q - \underline{q}^d) \\ &= ((\mathbf{L}_p + P) \otimes I)(q - \underline{q}^d) \\ &= (\mathbf{L}_f \otimes I)x. \end{aligned} \quad (7.16)$$

Let $x = (x_1^T, \dots, x_N^T)^T$, $y = (y_1^T, \dots, y_N^T)^T$, $\delta = (\delta_1^T, \dots, \delta_N^T)^T$, $\eta = (\eta_1^T, \dots, \eta_N^T)^T$. It follows from (7.15) that

$$\begin{aligned}\dot{x} &= y, \\ \dot{y} &= -\text{sig}(\delta)^{\alpha_1} - \text{sig}(\eta)^{\alpha_2}, \\ \dot{\delta} &= \eta - k_1 \text{sig}(\delta - z)^{\sigma_1}, \\ \dot{\eta} &= -(\mathbf{L}_f \otimes I) \text{sig}(\delta)^{\alpha_1} - (\mathbf{L}_f \otimes I) \text{sig}(\eta)^{\alpha_2} - k_2 \text{sig}(\delta - z)^{\sigma_2}.\end{aligned}\quad (7.17)$$

By taking $X = (\mathbf{L}_f \otimes I)x$ and $Y = (\mathbf{L}_f \otimes I)y$, one obtains

$$\begin{aligned}\dot{X} &= Y, \\ \dot{Y} &= -(\mathbf{L}_f \otimes I) \text{sig}(\delta)^{\alpha_1} - (\mathbf{L}_f \otimes I) \text{sig}(\eta)^{\alpha_2}, \\ \dot{\delta} &= \eta - k_1 \text{sig}(\delta - X)^{\sigma_1}, \\ \dot{\eta} &= -(\mathbf{L}_f \otimes I) \text{sig}(\delta)^{\alpha_1} - (\mathbf{L}_f \otimes I) \text{sig}(\eta)^{\alpha_2} - k_2 \text{sig}(\delta - X)^{\sigma_2}.\end{aligned}\quad (7.18)$$

Let $e_X = \delta - X$ and $e_Y = \eta - Y$. One has

$$\begin{aligned}\dot{X} &= Y, \\ \dot{Y} &= -(\mathbf{L}_f \otimes I)(\text{sig}(X + e_X)^{\alpha_1} + \text{sig}(Y + e_Y)^{\alpha_2}), \\ \dot{e}_X &= e_Y - k_1 \text{sig}(e_X)^{\sigma_1}, \\ \dot{e}_Y &= -k_2 \text{sig}(e_X)^{\sigma_2}.\end{aligned}\quad (7.19)$$

By Lemma 7.1, $(\mathbf{L}_f \otimes I)$ is a positive definite. Then, according to Lemma 7.3 and (7.63), system (7.19) is globally finite-time stable. Since $X = (\mathbf{L}_f \otimes I)x$, $Y = (\mathbf{L}_f \otimes I)y$, it follows from Lemma 7.3 that $x = 0$, $y = 0$, which indicates that the states of the followers will converge to that of the leader in finite time. This completes the proof.

Remark 7.2 Protocol (7.12) is distributed since it relies merely on the relative output measurements between neighboring agents. Obviously, protocol (7.12) may also solve the finite-time distributed tracking problem with a stationary leader, i.e., $p_{N+1}^d(0) = 0$. Also, it is worth noting that Lemma 7.3 plays an important role in the proof of this result.

Remark 7.3 In contrast to previous works on solving the finite-time tracking problem by using static relative full-state information, protocol (7.12) is designed based only on relative output measurements, which is thus more favorable in practical applications. It is also worth mentioning that the separation principle of traditional observer-based controllers for a single system still holds in solving the finite-time consensus tracking problem for multi-agent systems. Moreover, by using protocol (7.12), the finite-time consensus tracking problem can be transformed into the finite-time stability problem for a single system.

7.1.3 Designing Formation Protocols Without Velocity Measurement

In this subsection, the finite-time control protocol (7.12) is extended to solving the formation control problem.

Let $h = (h_1^T, h_2^T, \dots, h_N^T)^T$ be the desired constant formation configuration.

Definition 7.2 The finite-time formation control problem of system (7.1) is said to be solved if there exists $T > 0$, such that $\lim_{t \rightarrow T} [(q_i - h_i) - (q_j - h_j)] = 0$, $\lim_{t \rightarrow T} (p_i - p_{N+1}^d) = 0$, and $(q_i - h_i) = (q_j - h_j)$, $p_i = p_{N+1}^d$, $t > T$, for all $i, j = 1, \dots, N$.

To achieve this goal, the following finite-time formation control protocol is proposed:

$$\begin{aligned}\dot{\delta}_i &= \eta_i - k_1 \text{sig}(\delta_i - z_i)^{\sigma_1}, \\ \dot{\eta}_i &= \sum_{j \in \mathbf{N}_i} a_{ij}(u_i - u_j) - k_2 \text{sig}(\delta_i - z_i)^{\sigma_2}, \\ z_i &= \sum_{j \in \mathbf{N}_i} a_{ij}[(q_i - h_i) - (q_j - h_j)], \\ u_i &= -\text{sig}(\delta_i)^{\alpha_1} - \text{sig}(\eta_i)^{\alpha_2}, \quad i = 1, \dots, N.\end{aligned}\tag{7.20}$$

Theorem 7.3 Suppose that Assumptions 7.1 and 7.2 hold. Then, the finite-time formation control problem of system (7.1) is solved by dynamical output feedback protocol (7.20), if

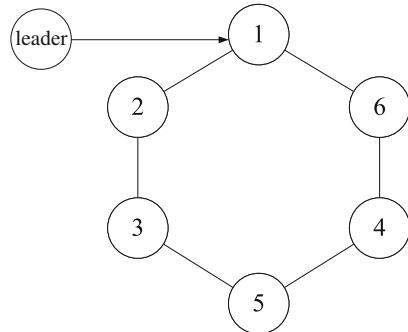
$$\begin{aligned}0 < \sigma_1 < 1, \quad \sigma_2 = 2\sigma_1 - 1, \\ \alpha_1 = 2\sigma_1 - 1, \quad \alpha_2 = (2\sigma_1 - 1)/\sigma_1.\end{aligned}\tag{7.21}$$

Proof Let $x_i = q_i - h_i - q_{N+1}^d$ and $y_i = p_i - p_{N+1}^d$, $i = 1, \dots, N$. Then, one has

$$\begin{aligned}\dot{x}_i &= y_i, \\ \dot{y}_i &= -\text{sig}(\delta_i)^{\alpha_1} - \text{sig}(\eta_i)^{\alpha_2}, \\ \dot{\delta}_i &= \eta_i - k_1 \text{sig}(\delta_i - z_i)^{\sigma_1}, \\ \dot{\eta}_i &= \sum_{j \in \mathbf{N}_i} a_{ij} \{-[\text{sig}(\delta_i)^{\alpha_1} - \text{sig}(\delta_j)^{\alpha_1}] \\ &\quad - [\text{sig}(\eta_i)^{\alpha_2} - \text{sig}(\eta_j)^{\alpha_2}]\} - k_2 \text{sig}(\delta_i - z_i)^{\sigma_2},\end{aligned}\tag{7.22}$$

where $z_i = \sum_{j \in \mathbf{N}_i} a_{ij}[(q_i - h_i) - (q_j - h_j)]$, $i = 1, \dots, N$. Let $x = (x_1^T, \dots, x_N^T)^T$, $y = (y_1^T, \dots, y_N^T)^T$, $\delta = (\delta_1^T, \dots, \delta_N^T)^T$, $\eta = (\eta_1^T, \dots, \eta_N^T)^T$, $z = (z_1^T, \dots, z_N^T)^T$, and $q = (q_1^T, \dots, q_N^T)^T$. It thus follows from (7.22) that

Fig. 7.1 Communication topology



$$\begin{aligned}
 \dot{x} &= y, \\
 \dot{y} &= -\text{sig}(\delta)^{\alpha_1} - \text{sig}(\eta)^{\alpha_2}, \\
 \dot{\delta} &= \eta - k_1 \text{sig}(\delta - (\mathbf{L}_f \otimes I)x)^{\sigma_1}, \\
 \dot{\eta} &= -(\mathbf{L}_f \otimes I)\text{sig}(\delta)^{\alpha_1} - (\mathbf{L}_f \otimes I)\text{sig}(\eta)^{\alpha_2} \\
 &\quad - k_2 \text{sig}(\delta - (\mathbf{L}_f \otimes I)x)^{\sigma_2}.
 \end{aligned} \tag{7.23}$$

Similarly to the proof of Theorem 7.5, x will converge to 0 in finite time. This indicates that the finite-time formation control problem of (7.1) is solved.

Example 7.1 Consider multi-agent system (7.1) with a communication topology as given by Fig. 7.1. Suppose that there is an active leader (7.2) in system (7.1), with $q_{N+1}^d(0) = (1, 2, 3)^T$ and $p_{N+1}^d(0) = (0.5, 0.5, 0.5)^T$. In simulations, let

$$\begin{aligned}
 h_1 &= (0.5, 0, -0.5)^T, \quad h_2 = (0.25, 0.6124, -0.25)^T, \\
 h_3 &= (-0.5, 0, 0.5)^T, \quad h_4 = (-0.25, 0.6124, 0.25)^T, \\
 h_5 &= (-0.25, -0.6124, 0.25)^T, \quad h_6 = (0.25, -0.6124, -0.25)^T.
 \end{aligned}$$

Choosing $\sigma_1 = 2/3$, $\sigma_2 = 1/3$, $\alpha_1 = 1/3$, and $\alpha_2 = 1/2$, the distributed control protocol (7.20) solves the finite-time formation problem. The simulation results are shown in Figs. 7.2 and 7.3. The six agents indeed achieve a regular hexagon shape with a separation of $\sqrt{2}/2$ in finite time.

7.1.4 Designing Saturated Consensus Protocols Without Velocity Measurement

In Sects. 7.1.1–7.1.3, we studied the distributed leader-following tracking problems on undirected graphs. The corresponding results can be extended to leaderless consensus with saturated protocols on directed graphs, as discussed below. We first

Fig. 7.2 Trajectories of the six agents in the three-dimensional space ($t \in [0, 10]$)

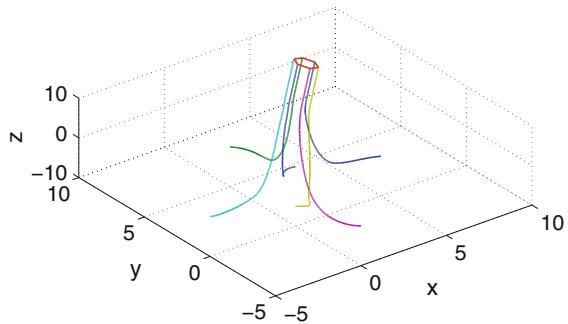
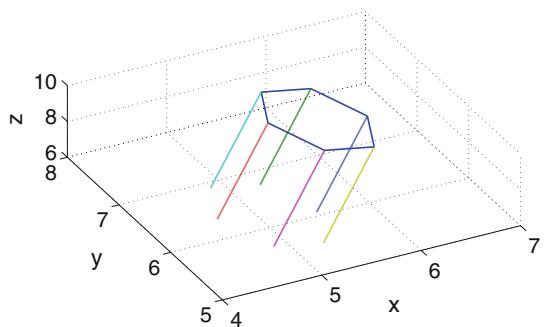


Fig. 7.3 Trajectories of the six agents in the three-dimensional space ($t \in [8, 10]$)



introduce the notion of detail-balanced graph [35–37]. A directed graph \mathbf{G} is said to satisfy the detail-balanced condition in weights if there exist some scalars $\theta_i > 0$, $i = 1, \dots, N$, such that $\theta_i a_{ij} = \theta_j a_{ji}$ for all $i, j = 1, \dots, N$.

Throughout this subsection, the communication topology satisfies the following assumption:

Assumption 7.3 The directed communication topology among agents is strongly connected and detail-balanced.

Lemma 7.4 Under Assumption 7.3, there exists a positive column vector $\theta = (\theta_1, \dots, \theta_N)^T$ such that $\theta_i a_{ij} = \theta_j a_{ji}$ for all $i, j = 1, \dots, N$. Moreover, matrix $\Theta \mathbf{L}$ is symmetric, where $\Theta = \text{diag}(\theta_1, \dots, \theta_N)$.

Consider a multi-agent system described by (7.1) without any leader. In what follows, the relative position measurements of other agents with respect to agent i are synthesized into a single signal as

$$z_i = \sum_{j=1}^N a_{ij}(q_i - q_j), \quad i = 1, \dots, N, \quad (7.24)$$

where $a_{ij} = 1$ if agent i can obtain information from agent j , but $a_{ij} = 0$ otherwise. Based on the above analysis, an observer-type consensus protocol is proposed as

$$\begin{aligned}\dot{\delta}_i &= \eta_i - \text{sig}(\delta_i - z_i)^{\sigma_1}, \\ \dot{\eta}_i &= \sum_{j=1}^N a_{ij}(u_i - u_j) - \text{sig}(\delta_i - z_i)^{\sigma_2}, \\ u_i &= -k_1 \tanh\{\text{sig}(\delta_i)^{\alpha_1}\} - k_2 \tanh\{\text{sig}(\eta_i)^{\alpha_2}\}, \quad i = 1, \dots, N,\end{aligned}\quad (7.25)$$

where k_1, k_2 are feedback gains and $\sigma_1, \sigma_2, \alpha_1, \alpha_2 \in \mathbf{R}$ are parameters to be determined. The function $\tanh(t) = \frac{e^t - e^{-t}}{e^t + e^{-t}}$ is saturated, therefore $\tanh(t)$ is bounded by $[-1, 1]$. The term $\sum_{j=1}^N a_{ij}(u_i - u_j)$ in (7.25) denotes the information exchanges between the protocol of agent i and those of its neighbors. Note that control protocol (7.25) is distributed, since it is based only on the relative information of neighboring agents.

Lemma 7.5 [34] Consider the following system:

$$\dot{\xi} = f(\xi) + \hat{f}(\xi), \quad f(0) = 0, \quad \xi \in \mathbf{R}^n, \quad (7.26)$$

where $\xi = (\xi_1, \dots, \xi_n)^T$, $f(\xi)$ is a continuous homogeneous vector field of degree $k < 0$ with respect to (r_1, \dots, r_n) , and $\hat{f}(\xi)$ satisfies $\hat{f}(0) = 0$. Assume $\xi = 0$ is an asymptotically stable equilibrium of the system $\dot{\xi} = f(\xi)$. Then, $\xi = 0$ is a locally finite-time stable equilibrium of the system if

$$\lim_{\varepsilon \rightarrow 0} \frac{\hat{f}_i(\varepsilon^{r_1}\xi_1, \dots, \varepsilon^{r_n}\xi_n)}{\varepsilon^{k+r_i}} = 0, \quad i = 1, \dots, n, \quad \forall \xi \neq 0.$$

Moreover, if the stable equilibrium $x = 0$ of the original system (7.26) is globally asymptotically stable, then $x = 0$ is a globally finite-time stable equilibrium of the system (7.26).

Theorem 7.4 Suppose that Assumption 7.3 holds. Then, the finite-time consensus problem for system (7.1) is solved by dynamical protocol (7.25), if

$$\begin{aligned}k_1 &> 0, \quad k_2 > 0, \\ \frac{1}{2} &< \sigma_1 < 1, \quad \sigma_2 = 2\sigma_1 - 1, \\ \alpha_1 &= 2\sigma_1 - 1, \quad \alpha_2 = \frac{2\sigma_1 - 1}{\sigma_1}.\end{aligned}\quad (7.27)$$

Proof Substituting (7.25) into (7.1) yields

$$\begin{aligned}\dot{q}_i &= p_i, \\ \dot{p}_i &= -k_1 \tanh\{\text{sig}(\delta_i)^{\alpha_1}\} - k_2 \tanh\{\text{sig}(\eta_i)^{\alpha_2}\}, \\ \dot{\delta}_i &= \eta_i - \text{sig}(\delta_i - z_i)^{\sigma_1},\end{aligned}$$

$$\begin{aligned}\dot{\eta}_i &= -k_1 \sum_{j=1}^N l_{ij} \tanh\{\text{sig}(\delta_j)^{\alpha_1}\} - k_2 \sum_{j=1}^N l_{ij} \tanh\{\text{sig}(\eta_j)^{\alpha_2}\} - \text{sig}(\delta_i - z_i)^{\sigma_2}, \\ z_i &= \sum_{j=1}^N a_{ij}(q_i - q_j), \quad i = 1, \dots, N.\end{aligned}\tag{7.28}$$

Let $q = (q_1^T, \dots, q_N^T)^T$, $p = (p_1^T, \dots, p_N^T)^T$, $z = (z_1^T, \dots, z_N^T)^T$, $\delta = (\delta_1^T, \dots, \delta_N^T)^T$ and $\eta = (\eta_1^T, \dots, \eta_N^T)^T$. It follows from (7.24) and (7.28) that

$$\begin{aligned}\dot{q} &= p, \\ \dot{p} &= -k_1 \tanh\{\text{sig}(\delta)^{\alpha_1}\} - k_2 \tanh\{\text{sig}(\eta)^{\alpha_2}\}, \\ \dot{\delta} &= \eta - \text{sig}(\delta - z)^{\sigma_1}, \\ \dot{\eta} &= -(\mathbf{L} \otimes I)[k_1 \tanh\{\text{sig}(\delta)^{\alpha_1}\} + k_2 \tanh\{\text{sig}(\eta)^{\alpha_2}\}] - \text{sig}(\delta - z)^{\sigma_2}, \\ z &= (\mathbf{L} \otimes I)q.\end{aligned}\tag{7.29}$$

Denote $e_x = \delta - (\mathbf{L} \otimes I)q$ and $e_y = \eta - (\mathbf{L} \otimes I)p$. One has

$$\begin{aligned}\dot{e}_x &= e_y - \text{sig}(e_x)^{\sigma_1}, \\ \dot{e}_y &= -\text{sig}(e_x)^{\sigma_2}.\end{aligned}\tag{7.30}$$

Consider a Lyapunov function candidate for system (7.30):

$$V_1 = \frac{1}{\sigma_2 + 1} |e_x|^{\sigma_2 + 1} + \frac{1}{2} \|e_y\|^2,\tag{7.31}$$

which is positive definite and radially unbounded since $\sigma_2 > 0$. The time derivative of V_1 along the trajectory of (7.30) is

$$\begin{aligned}\dot{V}_1 &= [e_y - \text{sig}(e_x)^{\sigma_1}]^T \text{sig}(e_x)^{\sigma_2} - e_y^T \text{sig}(e_x)^{\sigma_2} \\ &= -|e_x|^{\sigma_1 + \sigma_2},\end{aligned}\tag{7.32}$$

which is negative semi-definite. Now, note that $\dot{V}_1 = 0$ implies $e_x = 0$, which, in turn, implies $e_y = 0$ according to (7.30). By LaSalle's invariant set theorem, the zero solution of (7.30) is asymptotically stable. Since (7.30) is a homogeneous system of degree $\kappa = \frac{\sigma_2}{\sigma_1} - 1$ with dilation $(\frac{1}{\sigma_1}; 1)$, and condition (7.27) implies $\kappa < 0$, it follows from Lemma 7.5 that there exists $T_2 > 0$ such that $\lim_{t \rightarrow T_2} e_x = 0$ and $\lim_{t \rightarrow T_2} e_y = 0$, respectively, which means that $\delta = (\mathbf{L} \otimes I)q$ and $\eta = (\mathbf{L} \otimes I)p$, respectively, after a finite time T_2 . Therefore, one has

$$\begin{aligned}
u_i &= -k_1 \tanh\{\text{sig}(\delta_i)^{\alpha_1}\} - k_2 \tanh\{\text{sig}(\eta_i)^{\alpha_2}\} \\
&= -k_1 \tanh\left\{\text{sig}\left(\sum_{j=1}^N a_{ij}(q_i - q_j)\right)^{\alpha_1}\right\} \\
&\quad - k_2 \tanh\left\{\text{sig}\left(\sum_{j=1}^N a_{ij}(p_i - p_j)\right)^{\alpha_2}\right\}, \quad i = 1, \dots, N,
\end{aligned} \tag{7.33}$$

for $t > T_2$.

Next, we only need to prove the result that the above protocol (7.33) can guarantee the multi-agent systems (7.1) to achieve consensus under a directed graph. Let $M_d = I - \mathbf{1}r^T$, where $r^T = (r_1, \dots, r_N)$ satisfies $r^T \mathbf{L} = 0$ and $\sum_{i=1}^N r_i = 1$. Denote $x = (M_d \otimes I)q$ and $y = (M_d \otimes I)p$, respectively. Form (7.29) and (7.33), one obtains

$$\begin{aligned}
\dot{x} &= y, \\
\dot{y} &= -(M_d \otimes I) \left\{ k_1 \tanh\{\text{sig}[(\mathbf{L} \otimes I)x]^{\alpha_1}\} + k_2 \tanh\{\text{sig}[(\mathbf{L} \otimes I)y]^{\alpha_2}\} \right\}.
\end{aligned} \tag{7.34}$$

According to the algebraic properties of M_d , one has $x = 0$ and $y = 0$ if and only if $q_1 = q_2 = \dots = q_N$ and $p_1 = p_2 = \dots = p_N$, respectively. Therefore, the finite-time consensus problems can be solved if and only if x and y converge to zero in finite time, respectively.

Since $\tanh(\text{sig}(x)^\alpha) = \text{sig}(x)^\alpha + o(\text{sig}(x)^\alpha)$ around the origin [37], near the origin, system (7.34) can be rewritten as

$$\begin{aligned}
\dot{x} &= y, \\
\dot{y} &= -k_1(M_d \otimes I)\text{sig}[(\mathbf{L} \otimes I)x]^{\alpha_1} - k_2(M_d \otimes I)\text{sig}[(\mathbf{L} \otimes I)y]^{\alpha_2} + f(x, y),
\end{aligned} \tag{7.35}$$

where $f(x, y) = -k_1(M_d \otimes I)o(\text{sig}[(\mathbf{L} \otimes I)x]_1^\alpha) - k_2(M_d \otimes I)o(\text{sig}[(\mathbf{L} \otimes I)y]_2^\alpha)$.

It follows from Lemma 7.5 that the globally finite-time stability of the origin can be ensured if the following conditions hold:

1. The original system (7.35) is globally asymptotically stable;
2. The following system is also asymptotically stable and homogeneous:

$$\begin{aligned}
\dot{x} &= y, \\
\dot{y} &= -k_1(M_d \otimes I)\text{sig}[(\mathbf{L} \otimes I)x]^{\alpha_1} - k_2(M_d \otimes I)\text{sig}[(\mathbf{L} \otimes I)y]^{\alpha_2}; \tag{7.36}
\end{aligned}$$

3. $f(x, y)$ in (7.35) satisfies $\lim_{\varepsilon \rightarrow 0} \frac{f(\varepsilon^{r_1}x, \varepsilon^{r_2}y)}{\varepsilon^{\kappa+r_2}} = 0$, where, $\kappa < 0$, $r_1 > 0$ and $r_2 > 0$.

Consider a Lyapunov function candidate:

$$V_2 = k_1 \int_0^{(\mathbf{L} \otimes I)x} (\Theta \otimes I) \tanh[\text{sig}(s)^{\alpha_1}] ds + \frac{1}{2} y^T (\Theta \mathbf{L} \otimes I) y, \quad (7.37)$$

where $\Theta = \text{diag}(\theta_1, \dots, \theta_N) > 0$. Noting that $(\mathbf{L} \otimes I)x$ and $\text{sig}[(\mathbf{L} \otimes I)x]^{\alpha_1}$ have the same sign component-wise, one has $\int_0^{(\mathbf{L} \otimes I)x} (\Theta \otimes I) \tanh[\text{sig}(s)^{\alpha_1}] ds > 0$ for all $x \neq 0$. Moreover, it follows from $\Theta \mathbf{L} = (\Theta \mathbf{L})^T$ that $y^T (\Theta \mathbf{L} \otimes I) y > 0$ for all $y \neq 0$. Therefore, we obtain V_3 is positive definite. Then, the time derivative of V_2 along the trajectory of (7.34) is given by

$$\begin{aligned} \dot{V}_2 &= k_1 [(\mathbf{L} \otimes I)y]^T (\Theta \otimes I) \tanh\{\text{sig}[(\mathbf{L} \otimes I)x]^{\alpha_1}\} \\ &\quad - k_1 y^T (\Theta \mathbf{L} M_d \otimes I) \tanh\{\text{sig}[(\mathbf{L} \otimes I)x]^{\alpha_1}\} \\ &\quad - k_2 y^T (\Theta \mathbf{L} M_d \otimes I) \tanh\{\text{sig}[(\mathbf{L} \otimes I)y]^{\alpha_2}\} \\ &= -k_2 [(\mathbf{L} \otimes I)y]^T (\Theta \otimes I) \tanh\{\text{sig}[(\mathbf{L} \otimes I)y]^{\alpha_2}\} \leq 0. \end{aligned} \quad (7.38)$$

It can be seen that $\dot{V}_2 = 0$ together with (7.34) implies $x = y = 0$. It follows from LaSalle's invariant set theorem that the equilibrium of the closed-loop system (7.34) at the origin is globally asymptotically stable.

Next, consider system (7.36). A Lyapunov function candidate is chosen as

$$V_3 = \frac{k_1}{1 + \alpha_1} (\mathbf{1}^T \otimes I) (\Theta \otimes I) |(\mathbf{L} \otimes I)x|^{1+\alpha_1} + \frac{1}{2} y^T (\Theta \mathbf{L} \otimes I) y. \quad (7.39)$$

Taking the derivative of V_3 gives

$$\begin{aligned} \dot{V}_3 &= k_1 [(\mathbf{L} \otimes I)\dot{x}]^T (\Theta \otimes I) \text{sig}[(\mathbf{L} \otimes I)x]^{\alpha_1} + y^T (\Theta \mathbf{L} \otimes I) \dot{y} \\ &= k_1 [(\mathbf{L} \otimes I)y]^T (\Theta \otimes I) \text{sig}[(\mathbf{L} \otimes I)x]^{\alpha_1} \\ &\quad - y^T (\Theta \mathbf{L} M_d \otimes I) \{k_1 \text{sig}[(\mathbf{L} \otimes I)x]^{\alpha_1} + k_2 \text{sig}[(\mathbf{L} \otimes I)y]^{\alpha_2}\} \\ &= -k_2 [(\mathbf{L} \otimes I)y]^T (\Theta \otimes I) \text{sig}[(\mathbf{L} \otimes I)y]^{\alpha_2} \leq 0. \end{aligned} \quad (7.40)$$

Following a similar argument as above, one obtains that system (7.36) is globally asymptotically stable about its zero equilibrium. Also, by choosing $0 < \alpha_1 < 1$, $\alpha_2 = \frac{2\alpha_1}{1+\alpha_1}$, $\kappa = \frac{\alpha_1-1}{\alpha_1+1}$, $r_1 = \frac{2}{\alpha_1+1}$, and $r_2 = 1$, one can show that system (7.36) is also homogeneous of $\alpha_1 - 1 < 0$ with dilation $(\mathbf{1}^T, (1 + \alpha_1)\mathbf{1}^T)$.

Finally, similarly to the analysis in [34], one can easily get that $f(x, y)$ in (7.35) satisfies $\lim_{\varepsilon \rightarrow 0} \frac{f(\varepsilon^{r_1}x, \varepsilon^{r_2}y)}{\varepsilon^{\kappa+r_2}} = 0$ for all $(x^T, y^T)^T \neq 0$. This completes the proof.

7.1.5 Notes

The materials in Sect. 1.1 are mainly based on [14, 32]. Sections 1.1.1–1.1.4 are mainly based on [32, 38].

7.2 Distributed Finite-Time Containment Control of Second-Order Multi-agent Systems with Multiple Nonlinear Leaders

Suppose that the followers are governed by double-integrator dynamics described by

$$\dot{x}_i = v_i, \quad \dot{v}_i = u_i, \quad i = 1, \dots, N, \quad (7.41)$$

where x_i is the position, v_i is the velocity and u_i is the control input of the i th agent. The dynamics of the multiple leaders, labeled by $N+1, \dots, N+M$, are described by

$$\begin{aligned} \dot{x}_{N+i} &= v_{N+i}, \\ \dot{v}_{N+i} &= f_i(x_{N+i}, v_{N+i}, t), \quad i = 1, \dots, M, \end{aligned} \quad (7.42)$$

where $f_i(x_{N+i}, v_{N+i}, t)$ are leaders' accelerations. Assume $\|f_i(x_{N+i}, v_{N+i}, t)\|_\infty \leq c < \infty$.

Definition 7.3 For systems (7.41) and (7.42), the finite-time containment control problem is said to be solved if and only if there exist a distributed control protocol u_i and $T > 0$, such that the positions and velocities of the followers converge into the convex hulls $co(XL)$ and $co(VL)$, respectively, formed by those of the multiple leaders in finite time, where

$$\begin{aligned} co(XL) &= \left\{ \sum_{i=N+1}^{N+M} \vartheta_i x_i \mid \vartheta_i \geq 0, \sum_{i=N+1}^{N+M} \vartheta_i = 1 \right\}, \\ co(VL) &= \left\{ \sum_{i=N+1}^{N+M} \vartheta_i v_i \mid \vartheta_i \geq 0, \sum_{i=N+1}^{N+M} \vartheta_i = 1 \right\}. \end{aligned}$$

Assumption 7.4 The communication topology among the followers is undirected. For each follower, there exists at least one directed path from the leader to the follower.

Thus, \mathbf{L} can be rewritten as

$$\mathbf{L} = \begin{bmatrix} \mathbf{L}_f & \mathbf{L}_l \\ 0_{M \times N} & 0 \end{bmatrix}, \quad (7.43)$$

where $\mathbf{L}_l \in R^{N \times M}$ has at least one positive entry, \mathbf{L}_f is symmetric and positive definite, and $-L_1^{-1}\mathbf{L}_l\mathbf{1}_{M \times 1} = \mathbf{1}_{N \times 1}$ [18] under Assumption 7.4.

The main objective here is to design a distributed algorithm based on relative measurements of neighboring agents such that the states of the followers converge to a convex hull spanned by those of the multiple dynamic leaders in finite time.

7.2.1 Designing Finite-Time Containment Protocols Based on both Relative Position and Velocity Measurements

Consider the multi-agent system described by (7.41) and (7.42). The protocol based on relative position and velocity measurements is proposed as

$$u_i = -\alpha \text{sgn} \left\{ \sum_{j=1}^{N+M} a_{ij}(v_i - v_j) + \beta \text{sig} \left[\sum_{j=1}^{N+M} a_{ij}(x_i - x_j) \right]^{\frac{1}{2}} \right\}, \quad (7.44)$$

where α and β are positive constants to be determined, and a_{ij} is the (i, j) th entry of the adjacency matrix A . Throughout this subsection, solutions to the error systems are all understood in the sense of Filippov [39].

Theorem 7.5 Suppose that Assumption 7.4 holds. Then, the positions and velocities of the followers under protocol (7.44) will converge into the convex hulls $\text{co}(XL)$ and $\text{co}(VL)$ formed by those of the multiple leaders in finite time, respectively, if

$$-\alpha + c + \frac{\beta^2 \sqrt{Nn}}{2\lambda_1} < 0, \quad (7.45)$$

where λ_1 is the smallest eigenvalue of \mathbf{L}_f .

Proof By substituting (7.44) into (7.41), the closed-loop multi-agent system is transformed into

$$\begin{aligned} \dot{x}_i &= v_i, \\ \dot{v}_i &= -\alpha \text{sgn} \left\{ \sum_{j=1}^{N+M} a_{ij}(v_i - v_j) + \beta \text{sig} \left[\sum_{j=1}^{N+M} a_{ij}(x_i - x_j) \right]^{\frac{1}{2}} \right\}, \\ i &= 1, 2, \dots, N. \end{aligned} \quad (7.46)$$

Let $X = (x_1^T, \dots, x_N^T)^T$, $V = (v_1^T, \dots, x_N^T)^T$, $X_L = (x_{N+1}^T, \dots, x_{N+M}^T)^T$ and $V_L = (v_{N+1}^T, \dots, x_{N+M}^T)^T$. So, (7.46) can be rewritten in matrix form as

$$\begin{aligned}\dot{X} &= V, \\ \dot{V} &= -\alpha \text{sgn}(\mathbf{L}_f \otimes I)V + (\mathbf{L}_l \otimes I)V_L \\ &\quad + \beta \text{sig}[(\mathbf{L}_f \otimes I)X + (\mathbf{L}_l \otimes I)X_L]^{\frac{1}{2}}.\end{aligned}\quad (7.47)$$

Denote $\phi = (\phi_1^T, \dots, \phi_N^T)^T = -(\mathbf{L}_f^{-1} \mathbf{L}_l \otimes I)X_L$, $\varphi = (\varphi_1^T, \dots, \varphi_N^T)^T = -(\mathbf{L}_f^{-1} \mathbf{L}_l \otimes I)V_L$, $\tilde{X} = X - \phi$ and $\tilde{V} = V - \varphi$. One gets

$$\begin{aligned}\dot{\tilde{X}} &= \tilde{V}, \\ \dot{\tilde{V}} &= -\alpha \text{sgn}[(\mathbf{L}_f \otimes I)\tilde{V} + \beta \text{sig}[(\mathbf{L}_f \otimes I)\tilde{X}]^{\frac{1}{2}}] - F(x, v, t),\end{aligned}\quad (7.48)$$

where $F(x, v, t) = -(\mathbf{L}_f^{-1} \mathbf{L}_l \otimes I)[f_1^T, \dots, f_M^T]^T$. It follows from $-L_1^{-1} \mathbf{L}_l \mathbf{1}_{M \times 1} = \mathbf{1}_{N \times 1}$ and $\|f_i(x_{N+i}, v_{N+i}, t)\|_\infty \leq c < \infty$ that $\|F(x, v, t)\|_\infty \leq c < \infty$. Taking linear transformation $e_x = (\mathbf{L}_f \otimes I)\tilde{X}$ and $e_v = (\mathbf{L}_f \otimes I)\tilde{V}$, one has

$$\begin{aligned}\dot{e}_x &= e_v, \\ \dot{e}_v &= -\alpha(\mathbf{L}_f \otimes I)\text{sgn}[e_v + \beta \text{sig}(e_x)^{\frac{1}{2}}] - (\mathbf{L}_f \otimes I)F(x, v, t).\end{aligned}\quad (7.49)$$

According to $\mathbf{L}_f > 0$, one has $(\mathbf{L}_f \otimes I) > 0$. Let $S = e_v + \beta \text{sig}(e_x)^{\frac{1}{2}}$. Consider the following Lyapunov function

$$V_1 = \frac{1}{2} S^T (\mathbf{L}_f \otimes I)^{-1} S. \quad (7.50)$$

Differentiating the Lyapunov function V_1 along the system trajectory obtains

$$\begin{aligned}\dot{V}_1 &= S^T (\mathbf{L}_f \otimes I)^{-1} \dot{S} \\ &= S^T (\mathbf{L}_f \otimes I)^{-1} (\dot{e}_v + \beta \text{diag}(e_v)|e_x|^{-\frac{1}{2}}) \\ &= -\alpha S^T \text{sgn}(S) - S^T F(x, v, t) \\ &\quad + \beta S^T (\mathbf{L}_f \otimes I)^{-1} \text{diag}(e_v)|e_x|^{-\frac{1}{2}}.\end{aligned}\quad (7.51)$$

Checking the condition of $\dot{V}_1 < \text{const} < 0$ in a vicinity of each point on $S = 0$, and using $e_v = -\beta \text{sig}(e_x)^{\frac{1}{2}}$ verify that the 1-sliding-mode existence condition holds at each point except at the origin, if $-\alpha + c + \frac{\beta^2 \sqrt{Nn}}{2\lambda_1} < 0$, where λ_1 is the smallest eigenvalue of $(\mathbf{L}_f \otimes I)$. Then, it follows from the analysis of Proposition 1 in [40] that the trajectory of system (7.49) can not leave the origin. The same reasoning proves the Lyapunov stability of the origin, which implies that system (7.49) is globally finite-time stable if (7.45) is satisfied by appropriately choosing parameters α and β . Therefore, there exists $T > 0$ such that $\lim_{t \rightarrow T} e_x = \lim_{t \rightarrow T} e_v = 0$. By using $(\mathbf{L}_f \otimes I) > 0$ again, one obtains $\lim_{t \rightarrow T} \tilde{X} = \lim_{t \rightarrow T} \tilde{V} = 0$, namely, x_i and v_i will converge to ϕ_i and φ_i in finite time, respectively. It implies that the states of the

followers will move into the convex hulls $co(XL)$ and $co(XL)$, respectively. Therefore, the global finite-time containment control problem of multi-agent systems (7.41) and (7.42) is solved by protocol (7.44) using relative position and velocity measurements. This complete the proof.

Remark 7.4 Recall the existing results in [32], which studies the distributed finite-time tracking problem of multi-agent systems with one leader or multiple leaders without acceleration inputs. Notice that, in practice nonzero acceleration input is typically implemented on the leader to complete some predetermined tasks, for instance to avoid hazardous obstacles or to reach a desirable tracking target. This subsection considers the case of multiple dynamic leaders with nonzero acceleration inputs. The finite-time containment control protocol (7.44) is proposed and analyzed.

7.2.2 Designing Settling-Time Containment Protocols Based on both Relative Position and Velocity Measurements

In this subsection, the distributed settling-time tracking problem is investigated using relative position and velocity measurements, when acceleration measurement is not available.

First of all, the following useful lemmas are reviewed.

Lemma 7.6 (Young's inequality) *For every real numbers $a > 0, b > 0, c > 0, p > 1, q > 1$, with $\frac{1}{p} + \frac{1}{q} = 1$, the following inequality is satisfied:*

$$ab \leq c^p \frac{a^p}{p} + c^{-q} \frac{b^q}{q}.$$

Lemma 7.7 (Jensen's inequality) *For every real numbers $a \geq 0, b \geq 0$ and $0 < p < q$, the following inequality is satisfied:*

$$(a^q + b^q)^{\frac{1}{q}} \leq (a^p + b^p)^{\frac{1}{p}}.$$

Consider the multi-agent system described by (7.41) and (7.42). The protocol based on relative position and velocity measurements is proposed as

$$\begin{aligned} u_i &= -k_1 \operatorname{sgn} \left[\sum_{j=1}^{N+M} a_{ij}(x_i - x_j) \right] - k_2 \operatorname{sgn} \left[\sum_{j=1}^{N+M} a_{ij}(v_i - v_j) \right], \\ i &= 1, \dots, N, \end{aligned} \tag{7.52}$$

where k_1 and k_2 are positive constants to be determined, and a_{ij} is the (i, j) th entry of the adjacency matrix A . Throughout this subsection, solutions of the error systems should be understood in the sense of Filippov [39].

The following result provides a sufficient condition for the designed consensus tracking protocol (7.52).

Theorem 7.6 Suppose that Assumption 7.4 holds. Then, the positions and velocities of the followers under protocol (7.52) will converge into the convex hulls $\text{co}(XL)$ and $\text{co}(VL)$ formed by those of the multiple leaders in finite settling time respectively, if the following inequalities are fulfilled:

$$k_1 > \frac{l\theta}{3} (4(N-1)n)^{\frac{1}{3}} \sqrt{\frac{2(N-1)n}{\lambda_{\max}(\mathbf{L}_f)}}, \quad (7.53)$$

$$k_1 > k_2 + c, \quad (7.54)$$

$$k_2 > \frac{2\theta(N-1)n\lambda_{\min}(\mathbf{L}_f)}{3} \sqrt{\frac{2(N-1)n}{\lambda_{\max}(\mathbf{L}_f)}} + c, \quad (7.55)$$

with $\theta > 0$ and $l = \|\mathbf{L}_f^{-1}\|_\infty$. Moreover, using the Lyapunov function

$$V_1 = \left[k_1 \|e_x\|_1 + \frac{1}{2} e_v^T (\mathbf{L}_f^{-1} \otimes I) e_v \right]^{\frac{3}{2}} + \theta e_x^T (\mathbf{L}_f^{-1} \otimes I) e_v, \quad (7.56)$$

the settling-time estimation can be computed by

$$T_1 = \frac{3}{\gamma} V_1^{\frac{1}{3}}(e_x(0), e_v(0)), \quad (7.57)$$

with γ satisfying $\alpha_1^{\frac{3}{2}} - \gamma^{\frac{3}{2}}\beta_1 > 0$ and $\alpha_2^{\frac{3}{2}} - \gamma^{\frac{3}{2}}\beta_2 > 0$, where $\alpha_1 = \theta(k_1 - k_2 - c)$, $\alpha_2 = \frac{3}{2} \left(\frac{\lambda_{\max}(\mathbf{L}_f)}{2Nn} \right)^{\frac{1}{2}} (k_2 - c) - \theta(N-1)n\lambda_{\min}(\mathbf{L}_f)$, $\beta_1 = \sqrt{2}k_1^{\frac{3}{2}} + \theta \frac{2h^{\frac{3}{2}}}{3}$ and $\beta_2 = \theta \frac{(N-1)nl^3}{3h^3}$.

Proof By substituting (7.52) into (7.41), the closed-loop systems are obtained as

$$\begin{aligned} \dot{x}_i &= v_i, \\ \dot{v}_i &= -k_1 \text{sgn} \left[\sum_{j=1}^{N+M} a_{ij}(x_i - x_j) \right] - k_2 \text{sgn} \left[\sum_{j=1}^{N+M} a_{ij}(v_i - v_j) \right]. \end{aligned} \quad (7.58)$$

Denote $\phi = (\phi_1^T, \dots, \phi_N^T)^T = -(\mathbf{L}_f^{-1} \mathbf{L}_l \otimes I) X_L$, $\varphi = (\varphi_1^T, \dots, \varphi_N^T)^T = -(\mathbf{L}_f^{-1} \mathbf{L}_l \otimes I) V_L$. Let $\tilde{x}_i = x_i - \phi_i$ and $\tilde{v}_i = v_i - \varphi_i$. Denote $\tilde{X} = (\tilde{x}_1^T, \dots, \tilde{x}_{N-1}^T)^T$, $\tilde{V} = (\tilde{v}_1^T, \dots, \tilde{v}_{N-1}^T)^T$ and $F(x_N, v_N, t) = (\mathbf{L}_f^{-1} \mathbf{L}_l \otimes I) f(x_N, v_N, t)$. Then, (7.58) can be rewritten in a matrix form as

$$\begin{aligned}\dot{\tilde{X}} &= \tilde{V}, \\ \dot{\tilde{V}} &= -k_1 \text{sgn}[(\mathbf{L}_f \otimes I)\tilde{X}] - k_2 \text{sgn}[(\mathbf{L}_f \otimes I)\tilde{V}] - F(x_N, v_N, t),\end{aligned}\quad (7.59)$$

where $\mathbf{L}_f > 0$. Taking linear transformation $e_x = (\mathbf{L}_f \otimes I)\tilde{X}$ and $e_v = (\mathbf{L}_f \otimes I)\tilde{V}$, one obtains

$$\begin{aligned}\dot{e}_x &= e_v \\ \dot{e}_v &= -(\mathbf{L}_f \otimes I)[k_1 \text{sgn}(e_x) + k_2 \text{sgn}(e_v) + F(x_N, v_N, t)].\end{aligned}\quad (7.60)$$

Then, consider the Lyapunov function (7.56). First of all, it can be shown that V_1 is positive definite. According to Lemmas 7.6 and 7.7, one has

$$\begin{aligned}V_1 &= \left[k_1 \|e_x\|_1 + \frac{1}{2} e_v^T (\mathbf{L}_f^{-1} \otimes I) e_v \right]^{\frac{3}{2}} + \theta e_x^T (\mathbf{L}_f^{-1} \otimes I) e_v \\ &\geq (k_1 \|e_x\|_1)^{\frac{3}{2}} + \left[\frac{1}{2} e_v^T (\mathbf{L}_f^{-1} \otimes I) e_v \right]^{\frac{3}{2}} \\ &\quad - \theta \left(\frac{2h^{\frac{3}{2}}}{3} \sum_{i=1}^{N-1} \sum_{m=1}^n |e_{xim}|^{\frac{3}{2}} + \frac{1}{3h^3} \|(\mathbf{L}_f^{-1} \otimes I)e_v\|_3^3 \right) \\ &\geq \left(k_1^{\frac{3}{2}} - \frac{2\theta h^{\frac{3}{2}}}{3} \right) \left(\sum_{i=1}^{N-1} \sum_{m=1}^n |e_{xim}|^{\frac{3}{2}} \right) \\ &\quad + \left[\left(\frac{\lambda_{\max}(\mathbf{L}_f)}{2(N-1)n} \right)^{\frac{3}{2}} - \theta \frac{(N-1)nl^3}{3h^3} \right] \|e_v\|_1^3.\end{aligned}\quad (7.61)$$

Both coefficients, i.e., $k_1^{\frac{3}{2}} - \frac{2\theta h^{\frac{3}{2}}}{3}$ and $\left(\frac{\lambda_{\max}(\mathbf{L}_f)}{2(N-1)n} \right)^{\frac{3}{2}} - \theta \frac{(N-1)nl^3}{3h^3}$ in the above expression are positive if h is selected such that $l \left(\frac{\theta(N-1)n}{3} \right)^{\frac{1}{3}} \sqrt{\frac{2(N-1)n}{\lambda_{\max}(\mathbf{L}_f)}} < h < \left(\frac{3}{2\theta} \right)^{\frac{2}{3}} k_1$. Such an h exists if

$$k_1 > \frac{l\theta}{3} (4(N-1)n)^{\frac{1}{3}} \sqrt{\frac{2(N-1)n}{\lambda_{\max}(\mathbf{L}_f)}}.\quad (7.62)$$

Select $h = \varepsilon l \left(\frac{\theta(N-1)n}{3} \right)^{\frac{1}{3}} \sqrt{\frac{2(N-1)n}{\lambda_{\max}(\mathbf{L}_f)}} + (1-\varepsilon) \left(\frac{3}{2\theta} \right)^{\frac{2}{3}} k_1$, $0 < \varepsilon < 1$, which guarantees the positive definiteness of V_1 . Then, differentiating the Lyapunov function V_1 , along the trajectory of (7.60) gives

$$\begin{aligned}
\dot{V}_1 &= \frac{3}{2} \left[k_1 \|e_x\|_1 + \frac{1}{2} e_v^T (\mathbf{L}_f^{-1} \otimes I) e_v \right]^{\frac{1}{2}} \left[k_1 e_v^T \operatorname{sgn}(e_x) \right. \\
&\quad \left. - e_v^T \left(k_1 \operatorname{sgn}(e_x) + k_2 \operatorname{sgn}(e_v) + F \right) \right] \\
&\quad + \theta e_v^T (\mathbf{L}_f^{-1} \otimes I) e_v \\
&\quad - \theta e_x^T \left(k_1 \operatorname{sgn}(e_x) + k_2 \operatorname{sgn}(e_v) + F \right) \\
&= -\frac{3}{2} \left[k_1 \|e_x\|_1 + \frac{1}{2} e_v^T (\mathbf{L}_f^{-1} \otimes I) e_v \right]^{\frac{1}{2}} \\
&\quad \left(k_2 \|e_v\|_1 - e_v^T F \right) + \theta e_v^T (\mathbf{L}_f^{-1} \otimes I) e_v - \theta [k_1 \|e_x\|_1 \right. \\
&\quad \left. + k_2 e_x^T \operatorname{sgn}(e_v) + e_x^T F].
\end{aligned}$$

Note that the following inequalities hold for all t, e_x, e_v :

$$k_2 \|e_v\|_1 - e_v^T F > 0, \quad k_1 \|e_x\|_1 + k_2 e_x^T \operatorname{sgn}(e_v) + e_x^T F > 0,$$

if

$$k_2 > c, \quad k_1 > k_2 + c. \quad (7.63)$$

In this case, since $[k_1 \|e_x\|_1 + \frac{1}{2} e_v^T (\mathbf{L}_f^{-1} \otimes I) e_v]^{\frac{1}{2}} \geq \left(\frac{\lambda_{\max}(\mathbf{L}_f)}{2} \right)^{\frac{1}{2}} \|e_v\| \geq \left(\frac{\lambda_{\max}(\mathbf{L}_f)}{2(N-1)n} \right)^{\frac{1}{2}} \|e_v\|_1$ and $e_v^T (\mathbf{L}_f^{-1} \otimes I) e_v \leq \lambda_{\min}(\mathbf{L}_f) \|e_v\|^2 \leq (N-1)n \lambda_{\min}(\mathbf{L}_f) \|e_v\|_1^2$, it follows that

$$\begin{aligned}
\dot{V}_1 &\leq - \left[\frac{3}{2} \left(\frac{\lambda_{\max}(\mathbf{L}_f)}{2(N-1)n} \right)^{\frac{1}{2}} (k_2 - c) - \theta N n \lambda_{\min}(\mathbf{L}_f) \right] \\
&\quad \|e_v\|_1^2 - \theta \left(k_1 - k_2 - c \right) \|e_x\|_1.
\end{aligned} \quad (7.64)$$

Thus, it is clear that \dot{V}_1 is negative definite if

$$k_2 > \frac{2\theta(N-1)n\lambda_{\min}(\mathbf{L}_f)}{3} \sqrt{\frac{2(N-1)n}{\lambda_{\max}(\mathbf{L}_f)}} + c, \quad k_1 > k_2 + c.$$

Note that

$$\begin{aligned}
& \left[k_1 \|e_x\|_1 + \frac{1}{2} e_v^T (\mathbf{L}_f^{-1} \otimes I) e_v \right]^{\frac{3}{2}} \\
& \leq \sqrt{2} \left[(k_1 \|e_x\|_1)^{\frac{3}{2}} + \left(\frac{1}{2} e_v^T (\mathbf{L}_f^{-1} \otimes I) e_v \right)^{\frac{3}{2}} \right] \\
& \leq \sqrt{2} \left[(k_1 \|e_x\|_1)^{\frac{3}{2}} + \left(\frac{\lambda_{\min}(\mathbf{L}_f)}{2(N-1)n} \right)^{\frac{3}{2}} \|e_x\|_1^3 \right]. \tag{7.65}
\end{aligned}$$

One has

$$\begin{aligned}
V_1 &= \left[k_1 \|e_x\|_1 + \frac{1}{2} e_v^T (\mathbf{L}_f^{-1} \otimes I) e_v \right]^{\frac{3}{2}} \\
&\quad + \theta e_x^T (\mathbf{L}_f^{-1} \otimes I) e_v \\
&\leq \sqrt{2} \left[(k_1 \|e_x\|_1)^{\frac{3}{2}} + \left(\frac{1}{2} e_v^T (\mathbf{L}_f^{-1} \otimes I) e_v \right)^{\frac{3}{2}} \right] \\
&\quad + \theta \left(\frac{2h^{\frac{3}{2}}}{3} \sum_{i=1}^{N-1} \sum_{m=1}^n |e_{xim}|^{\frac{3}{2}} + \frac{1}{3h^3} \|(\mathbf{L}_f^{-1} \otimes I) e_v\|_3^3 \right) \\
&\leq \left(\sqrt{2} k_1^{\frac{3}{2}} + \theta \frac{2h^{\frac{3}{2}}}{3} \right) \|e_x\|_1^{\frac{3}{2}} \\
&\quad + \left[\left(\frac{\lambda_{\min}(\mathbf{L}_f)}{2(N-1)n} \right)^{\frac{3}{2}} + \theta \frac{Nnl^3}{3h^3} \right] \|e_v\|_1^3 \\
&= \beta_1 \|e_x\|_1^{\frac{3}{2}} + \beta_2 \|e_v\|_1^3, \tag{7.66}
\end{aligned}$$

where $\beta_1 = \sqrt{2} k_1^{\frac{3}{2}} + \theta \frac{2h^{\frac{3}{2}}}{3}$ and $\beta_2 = \theta \frac{(N-1)nl^3}{3h^3}$. Let $\alpha_1 = \theta(k_1 - k_2 - c)$ and $\alpha_2 = \frac{3}{2} \left(\frac{\lambda_{\max}(\mathbf{L}_f)}{2Nn} \right)^{\frac{1}{2}} (k_2 - c) - \theta(N-1)n\lambda_{\min}(\mathbf{L}_f)$. Recalling (7.64), it follows from Lemma 7.7 that

$$\begin{aligned}
\dot{V}_1 &\leq -\alpha_1 \|e_x\|_1 - \alpha_2 \|e_v\|_1^2 \\
&\leq -(\alpha_1^{\frac{3}{2}} \|e_x\|_1^{\frac{3}{2}} + \alpha_2^{\frac{3}{2}} \|e_v\|_1^3)^{\frac{2}{3}} \\
&\leq -\gamma V_1^{\frac{2}{3}}, \tag{7.67}
\end{aligned}$$

where γ satisfies $\alpha_1^{\frac{3}{2}} - \gamma^{\frac{3}{2}} \beta_1 > 0$ and $\alpha_2^{\frac{3}{2}} - \gamma^{\frac{3}{2}} \beta_2 > 0$. Thus, e_x and e_v will converge to the origin in finite time, which means that \tilde{x}_i and \tilde{v}_i will be zero after a finite time. Therefore, the distributed consensus tracking problem with a leader of non-zero acceleration can be solved in finite time. Finally, from (7.67), one has

$V_1(t) = \left(V_1(0)^{\frac{1}{3}} - \frac{\gamma}{3}t \right)^3$, which implies that the trajectory starting from the initial point $(e_x(0), e_y(0))$ will converge to the origin in a finite settling time less than T_1 computed by (7.57). This completes the proof.

7.2.3 Designing Observer-Based Protocols Without Velocity Measurement

Note that in Theorems 7.5 and 7.6, the controllers so designed are relied on both relative position and velocity measurements. However, the controllers (7.44) and (7.52) are too strict for the case where only relative position data can be obtained. Therefore, in this subsection, the distributed finite-time containment control problem for multi-agent systems with multiple leaders having bounded unknown inputs is further studied under the assumption that only relative position measurements are used to design the controller.

In what follows, the relative position measurements of neighboring agents are synthesized into a single signal, in the form of

$$y_i = \sum_{j=1}^{N+M} a_{ij}(x_i - x_j), \quad i = 1, \dots, N. \quad (7.68)$$

Based on the above relative position measurements y_i , a finite-time position-based protocol is proposed as

$$\begin{aligned} \dot{\xi}_i &= \eta_i - l_1 \text{sig}(\xi_i - y_i)^{\frac{1}{2}}, \\ \dot{\eta}_i &= -l_2 \text{sgn}(\xi_i - y_i), \\ u_i^1 &= -\alpha \text{sgn}[\eta_i + \beta \text{sig}(\xi_i)^{\frac{1}{2}}], \\ u_i^2 &= -k_1 \text{sgn}(\xi_i) - k_2 \text{sgn}(\eta_i), \quad i = 1, 2, \dots, N, \end{aligned} \quad (7.69)$$

where $l_1, l_2, \alpha, \beta, k_1, k_2$ are parameters to be determined. Note that the controller (7.69) is distributed, since it is based only on local information.

Lemma 7.8 [41] Consider the system

$$\begin{aligned} \dot{p} &= q - l_1 \text{sig}(p)^{\frac{1}{2}}, \\ \dot{q} &= -l_2 \text{sgn}(p) + f(p, q, t), \end{aligned} \quad (7.70)$$

where l_1, l_2 are positive constants and the bounded perturbation satisfies $\|f(p, q, t)\|_\infty \leq \varepsilon$. Then, p and q will converge to zero in finite time if there exists a symmetric and positive definite matrix P such that

$$A^T P + PA + \varepsilon^2 C^T C + PBB^T P = -Q < 0, \quad (7.71)$$

where $A = \begin{bmatrix} -1/2k_1 & 1/2 \\ -k_2 & 0 \end{bmatrix}$, $B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$, $C = [1 \ 0]$.

Theorem 7.7 Suppose that Assumption 7.4 holds. Then, the finite-time containment control problem for multi-agent system (7.41) with dynamic leaders (7.42) of unknown bounded inputs is solved by (7.69), if

$$l_2 > 2DNn(c + \alpha), \quad l_1^2 > l_2, \quad (7.72)$$

hold, and α, β satisfy (7.45), k_1, k_2 satisfy (7.53), (7.54), (7.55), respectively, where D is the maximal in-degree of the network.

Proof Let $X = (x_1^T, x_2^T, \dots, x_N^T)^T$, $X_L = (x_{N+1}^T, x_{N+2}^T, \dots, x_{N+M}^T)^T$, $V = (v_1^T, v_2^T, \dots, v_N^T)^T$, $Y = (y_1^T, y_2^T, \dots, y_N^T)^T$, $\xi = (\xi_1^T, \xi_2^T, \dots, \xi_N^T)^T$, $\eta = (\eta_1^T, \eta_2^T, \dots, \eta_N^T)^T$ and $U_s = (u_1^{sT}, u_2^{sT}, \dots, u_N^{sT})^T$, $s = 1, 2$. According to (7.41), (7.42), (7.68) and (7.69), one gets

$$\begin{aligned} \dot{X} &= V, \\ \dot{V} &= U_s, \quad Y = \mathbf{L}_f X + \mathbf{L}_l X_L, \quad s = 1, 2, \\ \dot{\xi} &= \eta - l_1 \text{sig}(\xi - Y)^{\frac{1}{2}}, \\ \dot{\eta} &= -l_2 \text{sgn}(\xi - Y), \\ U_1 &= -\alpha \text{sgn}[\eta + \beta \text{sig}(\xi)^{\frac{1}{2}}], \\ U_2 &= -k_1 \text{sgn}(\xi) - k_2 \text{sgn}(\eta). \end{aligned} \quad (7.73)$$

Denote $\tilde{X} = X + (\mathbf{L}_f^{-1} \mathbf{L}_l \otimes I) X_L$ and $\tilde{V} = V + (\mathbf{L}_f^{-1} \mathbf{L}_l \otimes I) V_L$. It follows from (7.73) that

$$\begin{aligned} \dot{\tilde{X}} &= \tilde{V}, \\ \dot{\tilde{V}} &= U_s - F(x, v, t), \quad s = 1, 2, \\ U_1 &= -\alpha \text{sgn}[\eta + \beta \text{sig}(\xi)^{\frac{1}{2}}], \\ U_2 &= -k_1 \text{sgn}(\xi) - k_2 \text{sgn}(\eta), \\ \dot{\xi} &= \eta - l_1 \text{sig}(\xi - Y)^{\frac{1}{2}}, \\ \dot{\eta} &= -l_2 \text{sgn}(\xi - Y), \end{aligned} \quad (7.74)$$

where $F(x, v, t) = (-\mathbf{L}_f^{-1} \mathbf{L}_l \otimes I)[f_1(x_{N+1}, v_{N+1}, t)^T, \dots, f_M(x_{N+M}, v_{N+M}, t)^T]^T$. Since $\|f_i(x_{N+i}, v_{N+i}, t)\|_\infty \leq c < +\infty$, one has $\|F(x, v, t)\|_\infty \leq c < +\infty$. Taking $e_x = (\mathbf{L}_f \otimes I)\tilde{X} = Y$ and $e_v = (\mathbf{L}_f \otimes I)\tilde{V}$, one obtains

$$\begin{aligned}
\dot{e_x} &= e_v, \\
\dot{e_v} &= (\mathbf{L}_f \otimes I)U_s - (\mathbf{L}_f \otimes I)F(x, v, t), \quad s = 1, 2, \\
U_1 &= -\alpha \operatorname{sgn}[\eta + \beta \operatorname{sig}(\xi)^{\frac{1}{2}}], \\
U_2 &= -k_1 \operatorname{sgn}(\xi) - k_2 \operatorname{sgn}(\eta), \\
\dot{\xi} &= \eta - l_1 \operatorname{sig}(\xi - e_x)^{\frac{1}{2}}, \\
\dot{\eta} &= -l_2 \operatorname{sgn}(\xi - e_x).
\end{aligned} \tag{7.75}$$

Now, let $\delta_x = \xi - e_x$ and $\delta_v = \eta - e_v$. Then, one has

$$\begin{aligned}
\dot{\delta_x} &= \delta_v - l_1 \operatorname{sig}(\delta_x)^{\frac{1}{2}}, \\
\dot{\delta_v} &= -l_2 \operatorname{sgn}(\delta_x) + (\mathbf{L}_f \otimes I)U_s + (\mathbf{L}_f \otimes I)F(x, v, t), \quad s = 1, 2, \\
U_1 &= -\alpha \operatorname{sgn}[\eta + \beta \operatorname{sig}(\xi)^{\frac{1}{2}}], \\
U_2 &= -k_1 \operatorname{sgn}(\xi) - k_2 \operatorname{sgn}(\eta).
\end{aligned} \tag{7.76}$$

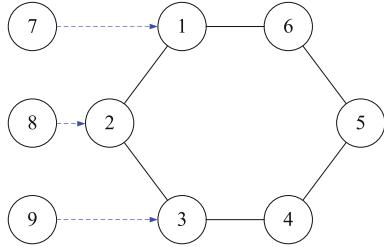
Since $\|(\mathbf{L}_f \otimes I)(U_1 + F(x, v, t))\|_\infty < 2DNn(c + \alpha)$ and $\|(\mathbf{L}_f \otimes I)(U_2 + F(x, v, t))\|_\infty < 2DNn(c + k_1 + k_2)$, where D is the maximal in-degree of network. It follows from Lemma 7.8 that there exists a finite time $T_1 > 0$, such that $\delta_x = 0$ and $\delta_v = 0$ when $t > T_1$, if there exists a positive definite matrix P such that (7.71) holds with $\varepsilon_1 = 2DNn(c + \alpha)$ and $\varepsilon_2 = 2DNn(c + k_1 + k_2)$. Furthermore, for a given bound $\varepsilon_1 = 2DNn(c + \alpha)$ and $\varepsilon_2 = 2DNn(c + k_1 + k_2)$ on leaders' inputs, it is easy to verify that (7.71) is feasible if $l_2 > \varepsilon_s$, $s = 1, 2$, and $l_1^2 > l_2$. In this case, δ_x and δ_v will converge in finite time to the origin, which means $\xi = (\mathbf{L}_f \otimes I)X + (\mathbf{L}_l \otimes I)X_L$ and $\eta = (\mathbf{L}_f \otimes I)V + (\mathbf{L}_l \otimes I)V_L$, respectively, when $t > T_1$. Thus, one has $\xi_i = \sum_{j=1}^{N+M} a_{ij}(x_i - x_j)$ and $\eta_i = \sum_{j=1}^{N+M} a_{ij}(v_i - v_j)$, respectively, when $t > T_1$. Therefore,

$$\begin{aligned}
u_i^1 &= -\alpha \operatorname{sgn}[\eta_i + \beta \operatorname{sig}(\xi_i)^{\frac{1}{2}}] \\
&= -\alpha \operatorname{sgn} \left\{ \sum_{j=1}^{N+M} a_{ij}(v_i - v_j) + \beta \operatorname{sig} \left[\sum_{j=1}^{N+M} a_{ij}(x_i - x_j) \right]^{\frac{1}{2}} \right\}, \\
i &= 1, 2, \dots, N,
\end{aligned} \tag{7.77}$$

and

$$\begin{aligned}
u_i^2 &= -k_1 \operatorname{sgn}(\xi_i) - k_2 \operatorname{sgn}(\eta_i) \\
&= -k_1 \operatorname{sgn} \left\{ \sum_{j=1}^{N+M} a_{ij}(x_i - x_j) \right\} \\
&\quad - k_2 \operatorname{sgn} \left\{ \sum_{j=1}^{N+M} a_{ij}(x_i - x_j) \right\}, \quad i = 1, 2, \dots, N,
\end{aligned} \tag{7.78}$$

Fig. 7.4 Communication topology



when $t > T_1$. This means that protocols (7.69) are identical to the control protocols (7.44) and (7.52), respectively, after the finite-time T_1 . So, it follows from Theorems 7.5 and 7.6 that the followers can track the convex hull formed by the multiple leaders in another finite-time $T_2 > 0$, if parameters α , β and k_1 , k_2 satisfy (7.45) and (7.53), (7.54), (7.55), respectively. To this end, the distributed finite-time containment control problem can be solved by using only relative position measurements in a total finite-time $T = T_1 + T_2$. This completes the proof.

Remark 7.5 Protocol (7.69) is distributed and based only on relative position measurements among neighboring agents. It is worth mentioning that Lemma 7.8 plays an important role in the derivation of this result. Furthermore, for the case of multiple leaders, protocol (7.69) can also solve the finite-time tracking control problem using only relative position measurements.

Example 7.2 Consider a network of 9 agents with an undirected topology among the six followers as shown in Fig. 7.4. Assume that there are three leaders described by the following dynamics:

$$\begin{aligned} \dot{x}_i &= v_{xi}, \\ \dot{y}_i &= v_{yi}, \\ \dot{v}_{xi} &= f_i(t), \\ \dot{v}_{yi} &= 0, \quad i = 7, 8, 9, \end{aligned} \tag{7.79}$$

where $f_7(t) = -\frac{1}{4}x_7 + \frac{1}{4}$, $f_8(t) = -\frac{1}{4}x_8 + \frac{3}{8}$ and $f_9(t) = -\frac{1}{4}x_9 + \frac{1}{2}$ with initial states $x_7(0) = 3$, $y_7(0) = 1$, $v_{x7}(0) = 0.5$, $v_{y7}(0) = 0.5$, $x_8(0) = 4$, $y_8(0) = 1.5$, $v_{x8}(0) = 0.5$, $v_{y8}(0) = 0.5$, $x_9(0) = 3$, $y_9(0) = 2$, and $v_{x9}(0) = 0.5$, $v_{y9}(0) = 0.5$. By choosing the controller parameters $\alpha = 10$, $\beta = 1$, $l_1 = 5$, and $l_2 = 10$, the conditions (7.45) and (7.27) in Theorem 7.7 are satisfied. Figures 7.5 and 7.6 show the positions and velocities of agents 1–9 using (7.69), respectively. It can be seen that the followers track the dynamic convex hull formed by the leaders at about $t = 4.5$, which verifies the effectiveness of the theoretical results.

Fig. 7.5 Trajectories of six agents with three dynamic leaders in the two-dimensional space

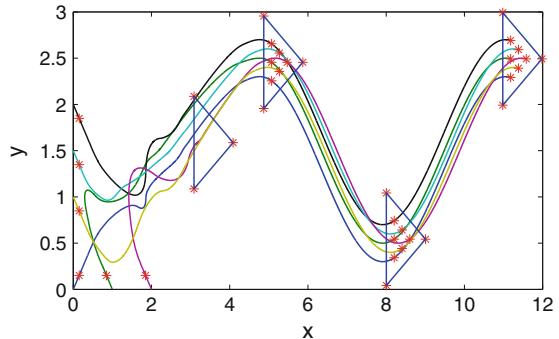
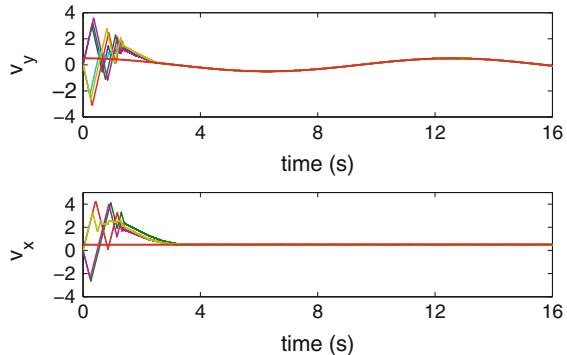


Fig. 7.6 Velocities of agents



7.2.4 Notes

The materials of Sect. 1.2 are mainly taken from [32, 42] with modifications. Sections 1.2.1–1.2.3 are mainly based on [42–44].

7.3 Distributed Finite-Time Tracking Control of Multiple Euler-Lagrange Systems

This section investigates the distributed finite-time tracking problem of networked agents with multiple Euler-Lagrange dynamics. To achieve finite-time tracking, a distributed finite-time protocol is first proposed based on both relative position and relative velocity measurements. By using tools from homogeneity theory, it is theoretically shown that the proposed protocol can guarantee finite-time tracking in the presence of control input constraints. Based on the state feedback analysis and with the aid of second-order sliding-mode observer approach, a new class of finite-time tracking protocols based only on relative position measurements is developed. It is proved that the multiple agents equipped with the designed protocols can track the

target location in finite time. Furthermore, a decentralized finite-time protocol based on a distributed estimator is proposed for solving the finite-time tracking problem with a dynamic leader.

Consider a network of multiple agents with Euler-Lagrange dynamics, whose behaviors are described by

$$M_i(q_i)\ddot{q}_i + C_i(q_i, \dot{q}_i)\dot{q}_i + G(q_i) = \tau_i, \quad i = 1, \dots, N, \quad (7.80)$$

where $q_i \in \mathbf{R}^n$ is a vector of generalized coordinates and is the measuring output, $M_i(q_i)$ represents the inertia matrix of rigid-body system i , $C_i(q_i, \dot{q}_i)\dot{q}_i$ includes the Coriolis and centrifugal forces and $G(q_i)$ denotes the gravitational force. For typical mechanical systems, the inertia matrices $M_i(q_i)$ are symmetric positive-definite with $0 < M_{\min} \leq M_i(q_i) \leq M_{\max}$, the Coriolis and centrifugal forces satisfy $\|C_i(q_i, \dot{q}_i)\| \leq k_c|\dot{q}_i|$, the gravitational force has $0 < \|G(q_i)\| \leq G_{\max}$, and $\dot{M}_i(q_i) - 2C_i(q_i, \dot{q}_i)$ are skew-symmetric matrices. Furthermore, when the accelerations in the mechanical system are bounded, the states of system (7.80) are also bounded.

7.3.1 Designing Finite-Time Tracking Protocols Based on both Relative Position and Velocity Measurement

It is assumed that there exists a stationary leader with constant position q_d . The control goal is to guarantee the followers' positions to track those of the leader in finite time. First, define the following auxiliary variables:

$$\begin{aligned} \zeta_i &= \sum_{k \in \mathbf{N}_i} a_{ik}(q_i - q_k), \\ v_i &= \sum_{k \in \mathbf{N}_i} a_{ik}(\dot{q}_i - \dot{q}_k), \quad i = 1, \dots, N. \end{aligned}$$

Then, a distributed finite-time tracking protocol is proposed for each follower i , as

$$\begin{aligned} \tau_i &= G(q_i) - \sum_{j \in \mathbf{N}_i} a_{ij}\{\tanh[\text{sig}(\zeta_i)^{\alpha_1}] - \tanh[\text{sig}(\zeta_j)^{\alpha_1}]\} \\ &\quad - \sum_{j \in \mathbf{N}_i} a_{ij}\{\tanh[\text{sig}(v_i)^{\alpha_2}] - \tanh[\text{sig}(v_j)^{\alpha_2}]\}, \\ i &= 1, \dots, N. \end{aligned} \quad (7.81)$$

Denote $x_i = q_i - q_d$ and $y_i = \dot{q}_i$, $i = 1, 2, \dots, N$. Submitting (7.81) into (7.80) gives

$$\begin{aligned}\dot{x}_i &= y_i, \\ \dot{y}_i &= -M_i(x_i + q_d)^{-1} \left\{ \sum_{j \in N_i} a_{ij} \{ \tanh[\text{sig}(\zeta_j)^{\alpha_1}] \right. \\ &\quad \left. - \tanh[\text{sig}(\zeta_j)^{\alpha_1}] \} + \sum_{j \in N_i} a_{ij} \{ \tanh[\text{sig}(v_i)^{\alpha_2}] \right. \\ &\quad \left. - \tanh[\text{sig}(v_j)^{\alpha_2}] \} + C_i(x_i + q_d, y_i) y_i \right\}.\end{aligned}\quad (7.82)$$

Since $\tanh(\text{sig}(x)^\alpha) = \text{sig}(x)^\alpha + o(\text{sig}(x)^\alpha)$ around the origin, near the origin, system (7.82) can be rewritten as

$$\begin{aligned}\dot{x}_i &= y_i, \\ \dot{y}_i &= M_i(q_d)^{-1} \left\{ - \sum_{j \in N_i} a_{ij} \{ \text{sig}(\zeta_i)^{\alpha_1} - \text{sig}(\zeta_j)^{\alpha_1} \} \right. \\ &\quad \left. - \sum_{j \in N_i} a_{ij} \{ \text{sig}(v_i)^{\alpha_2} - \text{sig}(v_j)^{\alpha_2} \} \right\} + f_i(x_i, y_i),\end{aligned}\quad (7.83)$$

where $f_i(x_i, v_i) = [M_i(x_i + q_d)^{-1} - M_i(q_d)^{-1}] \tau_i - C_i(x_i + q_d, y_i) y_i + M_i(q_d)^{-1} [o(\text{sig}(\zeta_i)^{\alpha_1}) + o(\text{sig}(v_i)^{\alpha_2})]$. Let $x = (x_1^T, \dots, x_N^T)^T$, $y = (y_1^T, \dots, y_N^T)^T$, $F(x, y) = (f_1^T, \dots, f_N^T)^T$, $\bar{q}_d = (\mathbf{1} \otimes I)q_d$, $C(x + \bar{q}_d, y) = \text{diag}(C_1(x_1 + q_d, y_1), \dots, C_N(x_N + q_d, y_N))$ and $M(x + \bar{q}_d)^{-1} = \text{diag}(M_1(x_1 + q_d)^{-1}, \dots, M_N(x_N + q_d)^{-1})$. Then, (7.82) and (7.83) can be rewritten in matrix form as

$$\begin{aligned}\dot{x} &= y, \\ \dot{y} &= -M(x + \bar{q}_d)^{-1} (\mathbf{L}_f \otimes I) \left\{ \tanh\{\text{sig}[(\mathbf{L}_f \otimes I)x]^{\alpha_1}\} \right. \\ &\quad \left. + \tanh\{\text{sig}[(\mathbf{L}_f \otimes I)y]^{\alpha_2}\} \right\} - M(x + \bar{q}_d)^{-1} C(x + \bar{q}_d, y) y,\end{aligned}\quad (7.84)$$

and

$$\begin{aligned}\dot{x} &= y, \\ \dot{y} &= -M(\bar{q}_d)^{-1} (\mathbf{L}_f \otimes I) \left\{ \text{sig}[(\mathbf{L}_f \otimes I)x]^{\alpha_1} + \text{sig}[(\mathbf{L}_f \otimes I)y]^{\alpha_2} \right\} + F(x, y),\end{aligned}\quad (7.85)$$

respectively. It follows from Lemma 7.5 that the globally finite-time stability of the origin can be ensured if the following conditions hold:

1. The original system (7.84) is globally asymptotically stable;
2. The following is also asymptotically stable and homogeneous,

$$\begin{aligned}\dot{x} &= y, \\ \dot{y} &= -M(\bar{q}_d)^{-1}(\mathbf{L}_f \otimes I) \left\{ \text{sig}[(\mathbf{L}_f \otimes I)x]^{\alpha_1} + \text{sig}[(\mathbf{L}_f \otimes I)y]^{\alpha_2} \right\};\end{aligned}\quad (7.86)$$

3. $F(x, y)$ in (7.85) satisfies $\lim_{\varepsilon \rightarrow 0} \frac{F(\varepsilon^{r_1}x, \varepsilon^{r_2}y)}{\varepsilon^{k+r_2}} = 0$, where, $k < 0, r_1 > 0$ and $r_2 > 0$.

Theorem 7.8 *Distributed tracking problem of multi-agent systems with Euler-Lagrange dynamics can be solved in finite time by using protocol (7.81) if the parameters are appropriately selected, as*

$$0 < \alpha_1 < 1, \quad \alpha_2 = \frac{2\alpha_1}{1 + \alpha_1}. \quad (7.87)$$

Moreover, the control input is bounded by $|\tau_i| \leq G_{\max} + 2d_G$.

Proof First, it is to show that system (7.84) is globally asymptotically stable about its zero point.

Let $X = (\mathbf{L}_f \otimes I)x$ and $Y = (\mathbf{L}_f \otimes I)y$. One has

$$\begin{aligned}\dot{X} &= Y, \\ \dot{Y} &= -(\mathbf{L}_f \otimes I)M(x + \bar{q}_d)^{-1}(\mathbf{L}_f \otimes I) \left\{ \tanh[\text{sig}(X)^{\alpha_1}] + \tanh[\text{sig}(Y)^{\alpha_2}] \right\} \\ &\quad - (\mathbf{L}_f \otimes I)M(x + \bar{q}_d)^{-1}C(x + \bar{q}_d, y)(\mathbf{L}_f \otimes I)^{-1}Y.\end{aligned}\quad (7.88)$$

Consider the Lyapunov function candidate

$$V_1 = \int_0^X \tanh[\text{sig}(s)^{\alpha_1}] ds + \frac{1}{2} Y^T (\mathbf{L}_f \otimes I)^{-1} M(x + \bar{q}_d) (\mathbf{L}_f \otimes I)^{-1} Y. \quad (7.89)$$

Since X and $\tanh[\text{sig}(X)^{\alpha_1}]$ have the same sign component-wise, one gets that $\int_0^X \tanh[\text{sig}(s)^{\alpha_1}] ds > 0$ for any $X \neq 0$. Furthermore, due to the fact $Y^T (\mathbf{L}_f \otimes I)^{-1} M(x + \bar{q}_d) (\mathbf{L}_f \otimes I)^{-1} Y > 0$ for any $Y \neq 0$, it can be concluded that V_1 is positive definite with respect to X and Y . Taking the derivative of V_1 gives

$$\begin{aligned}\dot{V}_1 &= \dot{X}^T \tanh[\text{sig}(X)^{\alpha_1}] + Y^T (\mathbf{L}_f \otimes I)^{-1} M(x + \bar{q}_d) (\mathbf{L}_f \otimes I)^{-1} \dot{Y} \\ &\quad + \frac{1}{2} Y^T (\mathbf{L}_f \otimes I)^{-1} \dot{M}(x + \bar{q}_d) (\mathbf{L}_f \otimes I)^{-1} Y \\ &= Y^T \tanh[\text{sig}(X)^{\alpha_1}] - Y^T \left\{ \tanh[\text{sig}(X)^{\alpha_1}] + \tanh[\text{sig}(Y)^{\alpha_2}] \right\} \\ &\quad - Y^T (\mathbf{L}_f \otimes I)^{-1} C(x + \bar{q}_d, y) (\mathbf{L}_f \otimes I)^{-1} Y \\ &\quad + \frac{1}{2} Y^T (\mathbf{L}_f \otimes I)^{-1} \dot{M}(x + \bar{q}_d) (\mathbf{L}_f \otimes I)^{-1} Y.\end{aligned}\quad (7.90)$$

Because $\dot{M}(x + \bar{q}_d) - 2C(x + \bar{q}_d, y)$ is skew-symmetric, one has $-Y^T(\mathbf{L}_f \otimes I)^{-1}C(x + \bar{q}_d, y)(\mathbf{L}_f \otimes I)^{-1}Y + \frac{1}{2}Y^T(\mathbf{L}_f \otimes I)^{-1}\dot{M}(x + \bar{q}_d)(\mathbf{L}_f \otimes I)^{-1}Y = \frac{1}{2}Y^T(\mathbf{L}_f \otimes I)^{-1}[\dot{M}(x + \bar{q}_d) - 2C(x + \bar{q}_d, y)](\mathbf{L}_f \otimes I)^{-1}Y = 0$. Thus,

$$\dot{V}_1 = -Y^T \tanh[\text{sig}(Y)^{\alpha_1}] \leq 0. \quad (7.91)$$

By noting that $\dot{V}_1 = 0$ implies $Y = 0$, it follows from (7.88) that $X = 0$. Therefore, by LaSalle's invariance principle, $\lim_{t \rightarrow +\infty} X = \lim_{t \rightarrow +\infty} Y = 0$. On the other hand, since $\mathbf{L}_f \otimes I$ is positive definite, one has $\lim_{t \rightarrow +\infty} x = \lim_{t \rightarrow +\infty} y = 0$, which means that the origin of system (7.84) is globally asymptotically stable.

Next, consider system (7.86). By taking the linear transformation $X = (\mathbf{L}_f \otimes I)x$ and $Y = (\mathbf{L}_f \otimes I)y$, one has that system (7.86) is globally asymptotically stable for the origin if and only if the following system is asymptotically stable for the origin.

$$\begin{aligned} \dot{X} &= Y, \\ \dot{Y} &= -(\mathbf{L}_f \otimes I)M(\bar{q}_d)^{-1}(\mathbf{L}_f \otimes I)\left\{\text{sig}[X]^{\alpha_1} + \text{sig}[Y]^{\alpha_2}\right\}. \end{aligned} \quad (7.92)$$

Choose the following Lyapunov function candidate:

$$V_2 = \frac{1}{1 + \alpha_1} \|X\|_1^{\alpha_1+1} + \frac{1}{2}Y^T(\mathbf{L}_f \otimes I)^{-1}M(\bar{q}_d)(\mathbf{L}_f \otimes I)^{-1}Y. \quad (7.93)$$

Taking the derivative of V_2 gives

$$\begin{aligned} \dot{V}_2 &= Y^T \text{sig}(X)^{\alpha_1} - Y^T \left\{\text{sig}[X]^{\alpha_1} + \text{sig}[Y]^{\alpha_2}\right\} \\ &= -\|Y\|_1^{1+\alpha_2} \leq 0. \end{aligned} \quad (7.94)$$

Following a similar argument as above, one can show that system (7.86) is globally asymptotically stable about its zero equilibrium. Also, by choosing $0 < \alpha_1 < 1$, $\alpha_2 = \frac{2\alpha_1}{1+\alpha_1}$, $k = \frac{\alpha_1-1}{1+\alpha_1}$, $r_1 = \frac{2}{1+\alpha_1}$ and $r_2 = 1$, one verifies that system (7.86) is also homogeneous of $\alpha_1 - 1 < 0$ with dilation $(\mathbf{1}^T, (1 + \alpha_1)\mathbf{1}^T)$.

Finally, by using a similar analysis as that in [34], one can easily show that $F(x, y)$ in (7.85) satisfies $\lim_{\varepsilon \rightarrow 0} \frac{F(\varepsilon^{r_1}x, \varepsilon^{r_2}y)}{\varepsilon^{k+r_2}} = 0$ for any $(x^T, y^T)^T \neq 0$.

Remark 7.6 Theorem 7.8 develops a finite-time tracking protocol in the presence of control input saturation. It is desirable in the case that the actuators embedded on each agent can only provide some bounded control forces. Moreover, it follows from mechanical properties that when the accelerations in the mechanical system (7.80) are bounded, its states are also bounded, which indicates that q_i and \dot{q}_i are bounded. Thus, by using protocol (7.81), one has that $|q_i| \leq \beta$ and $|\dot{q}_i| \leq \gamma$, respectively, where β and γ are some positive constants.

Remark 7.7 It follows from (7.81) that the protocol relies on both the one-hop neighboring agents and the two-hop ones. However, the research provided in the next subsection indicates that the requirement on two-hop's information can be removed by using an observer-based protocol.

Remark 7.8 In [45, 46], distributed finite-time containment and consensus problems have been studied, respectively. Compared with the results given in those references, this subsection presents a class of finite-time protocols to solve consensus tracking problems in the presence of input saturations, which is more important in practical applications. Then, a new class of observer-based algorithms are proposed and employed without the need for velocity information.

7.3.2 Designing Observer-Based Protocols Without Velocity Measurement

In practice, it is very hard or even impossible to obtain accurate measurement of the velocity \dot{q}_i of agent i . Since the information of \dot{q}_i is unavailable, the relative velocity information of neighboring agents cannot be used for feedback. It is thus more desirable to design a distributed finite-time protocol that is based only on the measurement of position q_i . Following the results given in Theorem 7.5, a finite-time consensus tracking algorithm is proposed below based only on the measurements of relative positions.

In this subsection, the following observer-based protocol is proposed:

$$\begin{aligned}\dot{\xi}_i &= \eta_i - c_1 \text{sig}(\xi_i - \zeta_i)^{\frac{1}{2}}, \\ \dot{\eta}_i &= -c_2 \text{sgn}(\xi_i - \zeta_i), \\ \tau_i &= G(q_i) - \sum_{j \in \mathbb{N}_i} l_{ij} \{\tanh[\text{sig}(\xi_j)^{\alpha_1}] + \tanh[\text{sig}(\eta_j)^{\alpha_2}]\},\end{aligned}\quad (7.95)$$

where ζ_i indicates the relative position measurements between its own and the neighbors, and c_1 and c_2 are parameters to be determined. Protocol (7.95) is distributed, since it is based only on local information.

Theorem 7.9 *Distributed tracking problem of multi-agent systems with Euler-Lagrange dynamics can be solved in finite time by using the protocol (7.95) if the parameters satisfy (7.87) and*

$$\begin{aligned}c_2 &> D, \\ c_1 &> 3 \sqrt{\frac{2}{c_2 - D}} (c_2 + D),\end{aligned}\quad (7.96)$$

where $D = \frac{d_{\mathbf{G}}}{M_{\min}}(2d_{\mathbf{G}} + k_c \gamma^2)$. Moreover, the control input is also bounded, satisfying $|\tau_i| \leq G_{\max} + 2d_{\mathbf{G}}$.

Proof Denote $x_i = q_i - q_d$ and $y_i = \dot{q}_i$. It follows from (7.80) and (7.95) that

$$\begin{aligned}\dot{x}_i &= y_i, \\ \dot{y}_i &= M_i(x_i + q_d)^{-1}\tau_i - M_i(x_i + q_d)^{-1}[G(q_i) + C_i(x_i + q_d, y_i)y_i], \\ \dot{\xi}_i &= \eta_i - c_1 \text{sig}(\xi_i - \zeta_i)^{\frac{1}{2}}, \\ \dot{\eta}_i &= -c_2 \text{sgn}(\xi_i - \zeta_i), \\ \tau_i &= G(q_i) - \sum_{j \in \mathbf{N}_i} l_{ij} \{\tanh[\text{sig}(\xi_j)^{\alpha_1}] + \tanh[\text{sig}(\eta_j)^{\alpha_2}]\}. \end{aligned} \quad (7.97)$$

Let $x = (x_1^T, \dots, x_N^T)^T$, $y = (y_1^T, \dots, y_N^T)^T$, $\xi = (\xi_1^T, \dots, \xi_N^T)^T$, $\eta = (\eta_1^T, \dots, \eta_N^T)^T$, $\zeta = (\zeta_1^T, \dots, \zeta_N^T)^T$, $\tau = (\tau_1^T, \dots, \tau_N^T)^T$, $\bar{q}_d = (\mathbf{1} \otimes I)q_d$, $C(x + \bar{q}_d, y) = \text{diag}(C_1(x_1 + q_d, y_1), \dots, C_N(x_N + q_d, y_N))$, $G(x + \bar{q}_d) = \text{diag}(G(x_1 + q_d), \dots, G_N(x_N + q_d))$ and $M(x + \bar{q}_d)^{-1} = \text{diag}(M_1(x_1 + q_d)^{-1}, \dots, M_N(x_N + q_d)^{-1})$. Then, (7.97) can be written in matrix form as

$$\begin{aligned}\dot{x} &= y, \\ \dot{y} &= M(x + \bar{q}_d)^{-1}\tau - M(x + \bar{q}_d)^{-1}[G(x + \bar{q}_d) + C(x + \bar{q}_d, y)y], \\ \dot{\xi} &= \eta - c_1 \text{sig}(\xi - \zeta)^{\frac{1}{2}}, \\ \dot{\eta} &= -c_2 \text{sgn}(\xi - \zeta), \\ \tau &= G(x + \bar{q}_d) - (\mathbf{L}_f \otimes I)\{\tanh[\text{sig}(\xi)^{\alpha_1}] + \tanh[\text{sig}(\eta)^{\alpha_2}]\}, \\ \zeta &= (\mathbf{L}_f \otimes I)x, \end{aligned} \quad (7.98)$$

By taking $X = (\mathbf{L}_f \otimes I)x$ and $Y = (\mathbf{L}_f \otimes I)y$, one obtains

$$\begin{aligned}\dot{X} &= Y, \\ \dot{Y} &= -(\mathbf{L}_f \otimes I)M(x + \bar{q}_d)^{-1}(\mathbf{L}_f \otimes I)\{\tanh[\text{sig}(\xi)^{\alpha_1}] + \tanh[\text{sig}(\eta)^{\alpha_2}]\}, \\ &\quad -(\mathbf{L}_f \otimes I)M(x + \bar{q}_d)^{-1}C(x + \bar{q}_d, y)(\mathbf{L}_f \otimes I)^{-1}Y \\ \dot{\xi} &= \eta - c_1 \text{sig}(\xi - X)^{\frac{1}{2}}, \\ \dot{\eta} &= -c_2 \text{sgn}(\xi - X). \end{aligned} \quad (7.99)$$

Let $e_X = \xi - X$ and $e_Y = \eta - Y$. One has

$$\begin{aligned}\dot{e}_X &= e_Y - c_1 \text{sig}(e_X)^{\frac{1}{2}}, \\ \dot{e}_Y &= -c_2 \text{sgn}(e_X) + g(q, \dot{q}, \xi, \eta), \end{aligned} \quad (7.100)$$

where, $g(q, \dot{q}, \xi, \eta) = (\mathbf{L}_f \otimes I)M(q)^{-1}\{(\mathbf{L}_f \otimes I)[\tanh(\text{sig}(\xi)^{\alpha_1}) + \tanh(\text{sig}(\eta)^{\alpha_2})] + C(q, \dot{q})\dot{q}\}$. Note that $\tau = G(q) - (\mathbf{L}_f \otimes I)\{\tanh[\text{sig}(\xi)^{\alpha_1}] + \tanh[\text{sig}(\eta_j)^{\alpha_2}]\}$ is bounded since $G(q)$ and $\tanh(\cdot)$ are bounded. Thus, $g(q, \dot{q}, \xi, \eta)$ is bounded with $D = \frac{d_G}{M_{\min}}(2d_G + k_c\gamma^2)$, i.e., $\|g(q, \dot{q}, \xi, \eta)\|_\infty \leq D$. By choosing c_1 and c_2 such that inequalities (7.96) holds, it follows from Theorem 1 in [47] that system (7.100) converges to the origin in a finite settling time less than $T_1 \leq \frac{\|\dot{e}_X\|_1}{c_2 - D}$, which implies that $(\xi, \eta) \rightarrow (X, Y)$ as $t < T_1$. Therefore, the estimated state $(\hat{\xi}, \hat{\eta})$ of the finite-time observer converges to the real states (X, Y) . Since $X = (\mathbf{L}_f \otimes I)x$ and $Y = (\mathbf{L}_f \otimes I)y$, one has

$$\hat{\xi}_i = \xi_i, \quad \eta_i = v_i, \quad t \geq T_1, \quad i = 1, \dots, N. \quad (7.101)$$

Thus,

$$\begin{aligned} \tau_i &= G(q_i) - \sum_{j \in \mathcal{N}_i} l_{ij}\{\tanh[\text{sig}(\xi_j)^{\alpha_1}] + \tanh[\text{sig}(\eta_j)^{\alpha_2}]\} \\ &= G(q_i) - \sum_{j \in \mathcal{N}_i} l_{ij}\{\tanh[\text{sig}(\zeta_j)^{\alpha_1}] + \tanh[\text{sig}(v_j)^{\alpha_2}]\}, \\ i &= 1, \dots, N, \quad t \geq T_1, \end{aligned} \quad (7.102)$$

where τ_i is given in protocol (7.81). Then, it follows from Theorem 7.5 that the position-based protocol (7.95) solves the finite-time consensus tracking problem with multiple Euler-Lagrange dynamics by choosing $0 < \alpha_1 < 1$ and $\alpha_2 = \frac{2\alpha_L}{1+\alpha_1}$. This completes the proof.

Remark 7.9 The proposed protocol in this subsection is partly motivated by the work in [46]. Different from the containment algorithms in [46], where it requires information from both the neighbors (one-hop neighbors) and the neighbors' neighbors (two-hop neighbors), the observer-based protocol proposed here is based only on the information from the neighbors (one-hop neighbors). In fact, variables ξ_i and η_i in (7.95), used for estimation, will converge to the real relative information ζ_i and v_i , respectively. Thus, the control inputs can be expressed as in (7.95). By this idea, the two-hop neighbor-based protocol can also be simplified to be a one-hop neighbor-based protocol. Therefore, (7.95) requires less information from both neighbors and itself.

Remark 7.10 Some idea from [47] have been employed here. When the acceleration in the mechanical system (7.80) are bounded, the constant D can be found as the double maximal possible acceleration of the system. Moreover, the estimation of the constant D does not depend on the nominal elasticity and the control terms. Such a state bound is true too, if the control input τ_i is bounded.

Remark 7.11 Protocol (7.95) is designed from the finite-time observer approach, which allows to design the distributed observer and the distributed controller separately, i.e., the separation principle is still satisfied for multi-agent systems.

7.3.3 Designing Observer-Based Protocols with a Dynamic Leader

In this subsection, the distributed finite-time tracking problem of multiple agents having Euler-Lagrange dynamics with a dynamic leader is investigated. It is assumed that both the generalized coordinates and generalized coordinate derivatives of the followers are available for all agents, but the leader's velocity and acceleration measurements are not available for the followers. Also, assume that the leader's acceleration \ddot{q}_d is bounded, $|\ddot{q}_d| < \delta$.

For follower dynamics (7.80), the following distributed finite-time cooperative tracking algorithm is proposed:

$$\begin{aligned} \dot{\hat{q}}_i &= \hat{p}_i, \quad \dot{\hat{p}}_i = \hat{r}_i, \quad \hat{r}_i = -\alpha_1 \text{sgn}[\hat{\eta}_i + \alpha_2 \text{sig}(\hat{\xi}_i)^{\frac{1}{2}}], \\ \dot{\hat{\xi}}_i &= \hat{\eta}_i - k_1 \text{sig} \left[\hat{\xi}_i - \sum_{j=0}^N a_{ij} (\hat{q}_i - \hat{q}_j) \right]^{\frac{1}{2}}, \\ \dot{\hat{\eta}}_i &= -k_2 \text{sgn} \left[\hat{\xi}_i - \sum_{j=0}^N a_{ij} (\hat{q}_i - \hat{q}_j) \right] + \sum_{j=0}^N a_{ij} (\hat{r}_i - \hat{r}_j), \\ \tau_i &= M_i(q_i)[G(q_i) + C_i(q_i, \dot{q}_i)\dot{q}_i + u_i], \\ u_i &= -l_1 \text{sgn}[(p_i - \hat{p}_i) + l_2 \text{sig}(q_i - \hat{q}_i)^{\frac{1}{2}}], \end{aligned} \quad (7.103)$$

where \hat{q}_i , \hat{p}_i , \hat{r}_i , $\hat{\xi}_i$ and $\hat{\eta}_i$, $i = 1, 2, \dots, N$, are the states of the dynamical protocol, with $\hat{q}_0 = q_d$ and $\hat{r}_0 = 0$, and $k_1, k_2, \alpha_1, \alpha_2, \mathbf{L}_f$ and l_2 are parameters to be determined.

Theorem 7.10 *Distributed tracking problem of multi-agent systems with Euler-Lagrange dynamics can be solved in finite time by using the protocol (7.103) if the parameters satisfy*

$$\begin{aligned} k_2 &> 2d_G\delta, \quad k_1^2 > k_2, \\ \alpha_1 &> \frac{\alpha_2^2}{2\lambda_1} \sqrt{N} + \delta, \quad l_1 > \frac{l_2^2}{2} + \delta. \end{aligned} \quad (7.104)$$

Proof Let $\tilde{q}_i = \hat{q}_i - q_d$ and $\tilde{p}_i = \hat{p}_i - \dot{q}_d$. One obtains that

$$\begin{aligned}
\dot{\tilde{q}}_i &= \tilde{p}_i, \\
\dot{\tilde{p}}_i &= -\alpha_1 \operatorname{sgn}[\widehat{\eta}_i + \alpha_2 \operatorname{sig}(\widehat{\xi}_i)^{\frac{1}{2}}] - \ddot{q}_d, \\
\dot{\widehat{\xi}}_i &= \widehat{\eta}_i - k_1 \operatorname{sig} \left[\widehat{\xi}_i - \sum_{j=0}^N a_{ij} (\tilde{q}_i - \tilde{q}_j) \right]^{\frac{1}{2}}, \\
\dot{\widehat{\eta}}_i &= -k_2 \operatorname{sgn} \left[\widehat{\xi}_i - \sum_{j=0}^N a_{ij} (\tilde{q}_i - \tilde{q}_j) \right] \\
&\quad - \alpha_1 \sum_{j=1}^N l_{ij} \operatorname{sgn}[\widehat{\eta}_i + \alpha_2 \operatorname{sig}(\widehat{\xi}_i)^{\frac{1}{2}}].
\end{aligned} \tag{7.105}$$

Let $\tilde{x}_i = \sum_{j=0}^N a_{ij} (\tilde{q}_i - \tilde{q}_j)$, $\tilde{y}_i = \sum_{j=0}^N a_{ij} (\tilde{p}_i - \tilde{p}_j)$, $\tilde{X} = (\tilde{x}_1^T, \dots, \tilde{x}_N^T)^T$, $\tilde{Y} = (\tilde{y}_1^T, \dots, \tilde{y}_N^T)^T$, $\widehat{\xi} = (\widehat{\xi}_1^T, \dots, \widehat{\xi}_N^T)^T$, $\widehat{\eta} = (\widehat{\eta}_1^T, \dots, \widehat{\eta}_N^T)^T$ and $Q_d = (\mathbf{1} \otimes I) q_d$. Then, (7.105) can be written in matrix form as

$$\begin{aligned}
\dot{\tilde{X}} &= \tilde{Y}, \\
\dot{\tilde{Y}} &= -\alpha_1 (\mathbf{L}_f \otimes I) \operatorname{sgn}[\eta + \alpha_2 \operatorname{sig}(\xi)^{\frac{1}{2}}] - (\mathbf{L}_f \otimes I) \ddot{Q}_d, \\
\dot{\widehat{\xi}} &= \widehat{\eta} - k_1 \operatorname{sig}(\widehat{\xi} - \tilde{X})^{\frac{1}{2}}, \\
\dot{\widehat{\eta}} &= -k_2 \operatorname{sgn}(\widehat{\xi} - \tilde{X}) - \alpha_1 (\mathbf{L}_f \otimes I) \operatorname{sgn}[\widehat{\eta} + \alpha_2 \operatorname{sig}(\widehat{\xi})^{\frac{1}{2}}].
\end{aligned} \tag{7.106}$$

Taking $e_1 = \widehat{\xi} - \tilde{X}$ and $e_2 = \widehat{\eta} - \tilde{Y}$ yields that

$$\begin{aligned}
\dot{e}_1 &= e_2 - k_1 \operatorname{sig}(e_1)^{\frac{1}{2}}, \\
\dot{e}_2 &= -k_2 \operatorname{sgn}(e_1) + (\mathbf{L}_f \otimes I) \ddot{Q}_d.
\end{aligned} \tag{7.107}$$

It follows from [41] that e_1 and e_2 will converge to the origin in finite time T_2 if $k_2 > 2d_G \delta$ and $k_1^2 > k_2$. Thus, one has $\widehat{\xi}_i = \sum_{j=0}^N a_{ij} (\tilde{q}_i - \tilde{q}_j)$ and $\widehat{\eta}_i = \sum_{j=0}^N a_{ij} (\tilde{p}_i - \tilde{p}_j)$ when $t > T_2$. Therefore, one has

$$\begin{aligned}
\dot{\tilde{q}}_i &= \tilde{p}_i, \\
\dot{\tilde{p}}_i &= -\alpha_1 \operatorname{sgn} \left[\sum_{j=0}^N a_{ij} (\tilde{p}_i - \tilde{p}_j) + \alpha_2 \operatorname{sig} \left(\sum_{j=0}^N a_{ij} (\tilde{q}_i - \tilde{q}_j) \right)^{\frac{1}{2}} \right] - \ddot{q}_d.
\end{aligned} \tag{7.108}$$

when $t > T_2$. Note that $\tilde{x}_i = \sum_{j=0}^N a_{ij} (\tilde{q}_i - \tilde{q}_j)$ and $\tilde{y}_i = \sum_{j=0}^N a_{ij} (\tilde{p}_i - \tilde{p}_j)$. When $t > T_2$, (7.108) can be written as

$$\begin{aligned}\dot{\tilde{X}} &= \tilde{Y}, \\ \dot{\tilde{Y}} &= -\alpha_1(\mathbf{L}_f \otimes I) \operatorname{sgn}[\tilde{Y} + \alpha_2 \operatorname{sig}(\tilde{X})^{\frac{1}{2}}] - (\mathbf{L}_f \otimes I) \ddot{Q}_d.\end{aligned}\tag{7.109}$$

Let $S = \tilde{Y} + \beta \operatorname{sig}(\tilde{X})^{\frac{1}{2}}$. Consider the Lyapunov function $V_3 = \frac{1}{2} S^T (\mathbf{L}_f \otimes I)^{-1} S$. Differentiating V_1 , along the trajectory of the system, one obtains

$$\begin{aligned}\dot{V}_3 &= -\alpha S^T \operatorname{sgn}(S) - S^T F(x, v, t) \\ &\quad + \frac{\beta}{2} S^T (\mathbf{L}_f \otimes I)^{-1} \operatorname{diag}(\tilde{Y}) |\tilde{X}|^{-\frac{1}{2}}.\end{aligned}\tag{7.110}$$

Checking the condition of $\dot{V}_3 < \text{const} < 0$ in a vicinity of each point on $S = 0$, using $\tilde{Y} = -\alpha_2 \operatorname{sig}(\tilde{X})^{\frac{1}{2}}$, one has that the 1-sliding-mode existence condition holds at each point except the origin, if $\alpha_1 > \frac{\alpha_2^2}{2\lambda_1} \sqrt{N} + \delta$, where λ_1 is the smallest eigenvalue of \mathbf{L}_f . Then, the trajectories of (7.109) inevitably hit the curve $S = 0$. Indeed, each trajectory starting from $S > 0$ will arrive at the semi-axis with $e_x = 0$ and $e_v < 0$, by ensuring $\operatorname{sgn}(S) = 1$. Thus, on the way it inevitably hits the curve $S = 0$. The same is true for the trajectory starting from $S < 0$. At that moment, the trajectory slides along the curve $S = 0$ towards the origin and reaches it in finite time. According to the analysis of Proposition 1 in [40], there exists a disk Θ_ε centered at the origin, such that every trajectory starting from Θ_ε will come to the origin in finite time, with the convergence time being uniformly bounded in Θ_ε . No trajectory starting from the origin can leave Θ_ε . Since ε can be taken arbitrarily small, the trajectory of system (7.109) will not leave the origin. Thus, the Lyapunov stability of the origin can be guaranteed, which implies that system (7.109) is globally finite-time stable. Therefore, there exists a finite time T_3 such that $\tilde{X} = \tilde{Y} = 0$ when $t > T_3$. It means that $\hat{q}_i = q_d$ and $\hat{p}_i = \dot{q}_d$ when $t > T_3$. Next, substituting τ_i and u_i in (7.103) to (7.80) yields

$$\ddot{q}_i = -l_1 \operatorname{sgn}[(p_i - \hat{p}_i) + l_2 \operatorname{sig}(q_i - \hat{q}_i)^{\frac{1}{2}}].\tag{7.111}$$

Since $\hat{q}_i = q_d$ and $\hat{p}_i = \dot{q}_d$ after the finite time T_3 , with $q_{ei} = q_i - q_d$ and $p_{ei} = p_i - \dot{q}_d$, one has the error system

$$\ddot{q}_{ei} = -l_1 \operatorname{sgn}[p_{ei} + l_2 \operatorname{sig}(q_{ei})^{\frac{1}{2}}] - \ddot{q}_d.\tag{7.112}$$

It follows from Proposition 1 in [40] again that q_{ei} and p_{ei} will converge to the origin after a finite time T_4 , if $l_1 > \frac{l_2^2}{2} + \delta$. Thus, protocol (7.103) solves the distributed finite-time tracking problem. This completes the proof.

Remark 7.12 It is worth mentioning that the protocol (7.103) is designed from the second-order sliding-mode control approach. Firstly, by using a distributed finite-time estimator, one can obtain the leader's position and velocity in finite time. Then, based on the separation principle in sliding-mode control theory, a decentralized controller is proposed for the networked agents with Euler-Lagrange dynamics. Thus, (7.103) is a decentralized finite-time controller with a distributed estimator.

Remark 7.13 Protocol (7.103) is based only on the leader's position, without using leader's velocity and acceleration measurements. Compared with protocols (7.81) and (7.95), which have solved the distributed finite-time tracking problems with a stationary leader, here (7.103) is designed to guarantee the networked agents with Euler-Lagrange dynamics to track a dynamic leader in finite time.

Example 7.3 Consider the multi-agent system (7.80) with multiple two-link manipulators as shown in Fig. 7.7, whose dynamics can be written explicitly as

$$\begin{pmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{pmatrix} \begin{pmatrix} \ddot{q}_{i1} \\ \ddot{q}_{i2} \end{pmatrix} + \begin{pmatrix} -h\dot{q}_{i2} & -h(\dot{q}_{i1} + \dot{q}_{i2}) \\ h\dot{q}_{i1} & 0 \end{pmatrix} \begin{pmatrix} \dot{q}_{i1} \\ \dot{q}_{i2} \end{pmatrix} = \begin{pmatrix} \tau_{i1} \\ \tau_{i2} \end{pmatrix},$$

where $H_{11} = a_1 + 2a_3\cos(q_{i2}) + 2a_4\sin(q_{i2})$, $H_{12} = H_{21} = a_2 + a_3\cos(q_{i2}) + a_4\sin(q_{i2})$, $H_{22} = a_2$ and $h = a_3\sin(q_2) - a_4\cos(q_2)$ with $a_1 = I_1 + m_1l_{c1}^2 + I_e + m_el_{ce}^2 + m_el_1^2$, $a_2 = I_e + m_el_{ce}^2$, $a_3 = m_el_1l_{ce}\cos(\delta_e)$ and $a_4 = m_el_1l_{ce}\sin(\delta_e)$. In simulations, set $m_1 = 1$, $l_1 = 1$, $m_e = 2$, $\delta_e = 30^\circ$, $I_1 = 0.12$, $l_{c1} = 0.5$, $I_e = 0.25$ and $l_{ce} = 0.6$. The robots, initially at rest $q_{11}(0) = 0^\circ$, $q_{12}(0) = 60^\circ$, $q_{21}(0) = 90^\circ$, $q_{22}(0) = -15^\circ$, $q_{31}(0) = 24^\circ$, $q_{32}(0) = 90^\circ$ and $q_{41}(0) = 96^\circ$, $q_{42}(0) = 0^\circ$, is commanded a step to $q_{d1} = 60^\circ$, $q_{d2} = 30^\circ$. The communication topology is visualized by Fig. 7.8. The corresponding transient positions and velocities are plotted in Figs. 7.9 and 7.10, with $\alpha_1 = 1/2$, $\alpha_2 = 2/3$. It can be seen that indeed the protocol (7.81) solves finite-time position tracking problem. Furthermore, it follows from Fig. 7.11 that the saturated control torques are bounded by 6, which illustrates the theoretical results.

Fig. 7.7 A two-link manipulator

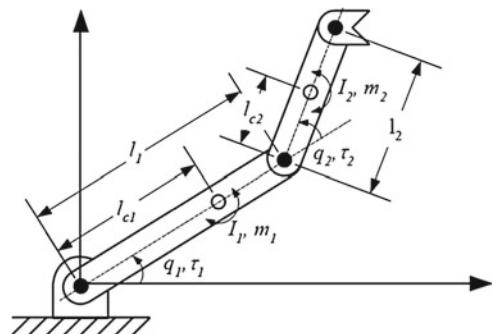


Fig. 7.8 Communication topology

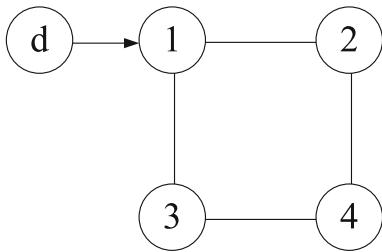


Fig. 7.9 Positions of all agents under protocol (7.81)

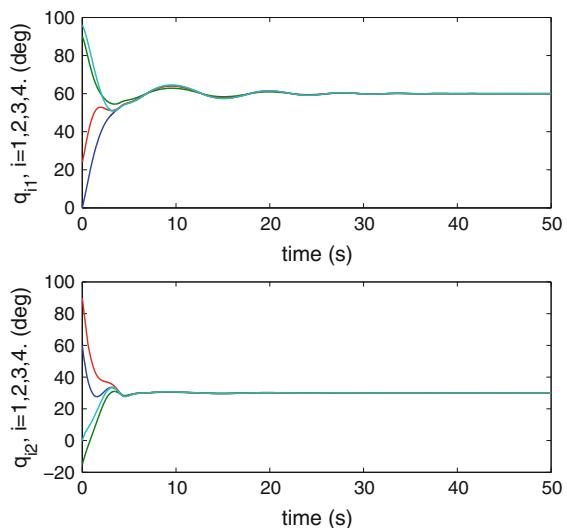


Fig. 7.10 Velocities of all agents under protocol (7.81)

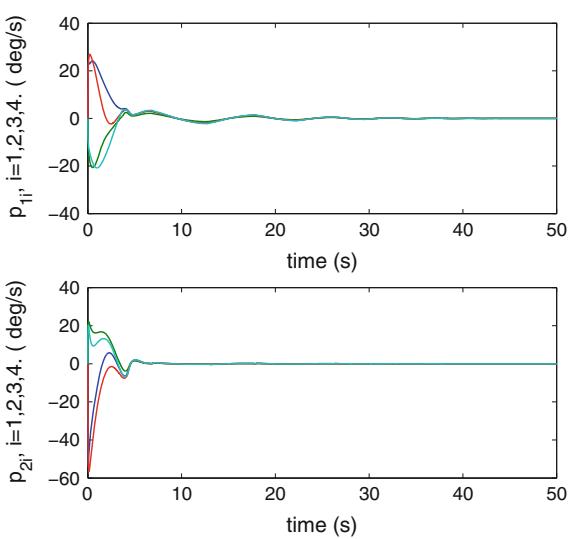
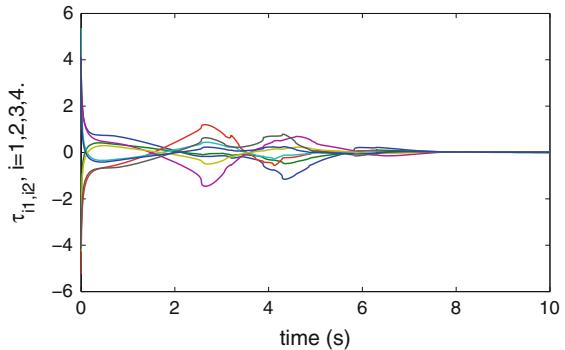


Fig. 7.11 Control torques under protocol (7.81)



Example 7.4 Consider the multi-agent system (7.80) with multiple two-link manipulators and choose identical parameters and the same communication topology as in the above example. The Simulation results are shown in Figs. 7.12 and 7.13, respectively, by using the observer-based protocol (7.95) with $c_1 = 20$, $c_2 = 50$. It can be seen that the distributed tracking problem without velocity measurements can be solved in finite time. Furthermore, the control torques shown in Fig. 7.14 are also bounded by 6, which illustrates the theoretical results. It is worth mentioning that the chattering phenomenon in Fig. 7.14 is due to the non-smooth control law in (7.95). How to reduce the chattering phenomenon is a further research topic.

Fig. 7.12 Positions of all agents under protocol (7.95)

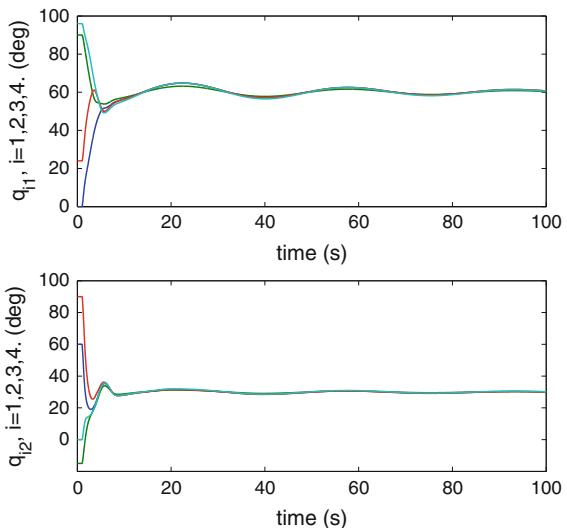


Fig. 7.13 Velocities of all agents under protocol (7.95)

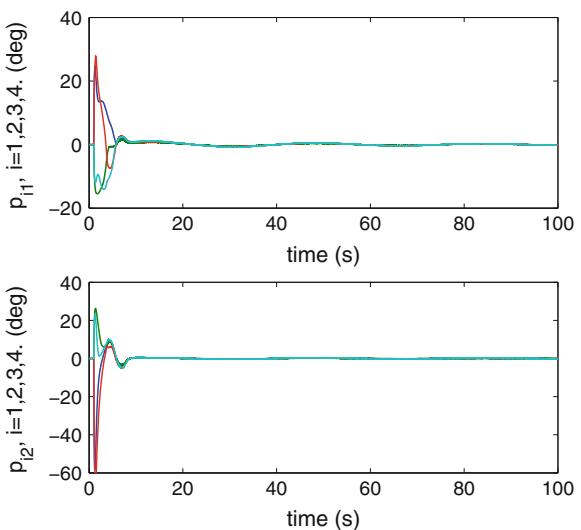
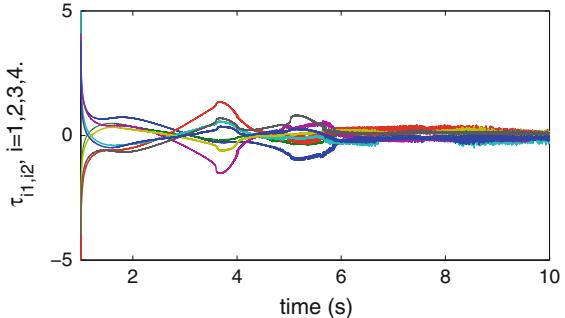


Fig. 7.14 Control torques under protocol (7.95)



7.3.4 Notes

The materials of Sect. 1.3 are mainly taken from [32, 42, 43] with modifications. Sections 1.2.1–1.2.3 are mainly based on [48].

References

1. Hong, Y., Chen, G., Bushnell, L.: Distributed observers design for leader-following control of multi-agent networks. *Automatica* **44**(3), 846–850 (2008)
2. Ji, M., Ferrari-Trecate, G., Egerstedt, M., Buffa, A.Y.: Containment control in mobile networks. *IEEE Trans. Autom. Control* **53**(8), 1972–1975 (2008)
3. Li, Z., Duan, Z., Chen, G., Huang, L.: Consensus of multi-agent systems and synchronization of complex networks: a unified viewpoint. *IEEE Trans. Circuits Syst. I: Regul. Pap.* **57**(1), 213–224 (2010)

4. Ren, W., Beard, R., Atkins, E.M.: Information consensus in multivehicle cooperative control. *IEEE Control Syst. Mag.* **27**(20), 71–82 (2007)
5. Jadbabaie, A., Lin, J., Morse, A.S.: Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Trans. Autom. Control* **48**(6), 988–1001 (2003)
6. Olfati-Saber, R., Murray, R.: Consensus problems in networks of agents with switching topology and time-delays. *IEEE Trans. Autom. Control* **49**(9), 1520–1533 (2004)
7. Vicsek, T., Czirók, A., Ben-Jacob, E., Cohen, I., Shochet, O.: Novel type of phase transition in a system of self-driven particles. *Phys. Rev. Lett.* **75**(6), 1226–1229 (1995)
8. Cao, M., Morse, A.S., Anderson, B.D.O.: Agreeing asynchronously. *IEEE Trans. Autom. Control* **53**(8), 1826–1838 (2008)
9. Ren, W., Beard, R.: Consensus seeking in multi-agent systems under dynamically changing interaction topologies. *IEEE Trans. Autom. Control* **50**(5), 655–661 (2005)
10. Ren, W., Beard, R.: *Distributed Consensus in Multi-Vehicle Cooperative Control*. Springer, London (2008)
11. Xiao, F., Wang, L., Chen, J., Gao, Y.: Finite-time formation control for multi-agent systems. *Automatica* **45**(11), 2605–2611 (2009)
12. Yu, W., Chen, G., Cao, M.: Some necessary and sufficient conditions for second-order consensus in multi-agent dynamical systems. *Automatica* **46**(6), 1089–1095 (2007)
13. Wen, G., Duan, Z., Yu, W., Chen, G.: Consensus in multi-agent systems with communication constraints. *Int. J. Robust Nonlinear Control* **22**(1), 170–182 (2012)
14. Zhao, Y., Duan, Z., Wen, G., Chen, G.: Distributed H_∞ consensus of multi-agent systems: a performance region-based approach. *Int. J. Control* **85**(3), 332–341 (2012)
15. Cao, Y., Ren, W.: Distributed coordinated tracking with reduced interaction via a variable structure approach. *IEEE Trans. Autom. Control* **57**(1), 33–48 (2012)
16. Ren, W.: Multi-vehicle consensus with a time-varying reference state. *Syst. Control Lett.* **56** (7–8), 474–483 (2007)
17. Li, Z., Liu, X., Ren, W., Xie, L.: Distributed tracking control for linear multi-agent systems with a leader of bounded unknown input. *IEEE Trans. Autom. Control* **58**(2), 518–523 (2013)
18. Cao, Y., Stuart, D., Ren, W., Meng, Z.: Distributed containment control for multiple autonomous vehicles with double-integrator dynamics: algorithms and experiments. *IEEE Trans. Control Syst. Technol.* **19**(4), 929–938 (2011)
19. Lou, Y., Hong, Y.: Multi-leader set coordination of multi-agent systems with random switching topologies. In: Proceedings of the 49th IEEE Conference on Decision and Control, pp. 3820–3825 (2010)
20. Notarstefano, G., Egerstedt, M., Haque, M.: Containment in leader-follower networks with switching communication topologies. *Automatica* **47**(5), 1035–1040 (2011)
21. Mei, J., Ren, W., Ma, G.: Containment control for multiple Euler-Lagrange systems with parametric uncertainties in directed networks. In: Proceedings of the 2011 American Control Conference, pp. 2186–2191 (2011)
22. Mei, J., Ren, W., Ma, G.: Distributed containment control for multiple nonlinear systems with identical dynamics. In: Proceedings of the 30th Chinese Control Conference, pp. 6544–6549 (2011)
23. Kim, Y., Mesbahi, M.: On maximizing the second smallest eigenvalue of a state-dependent graph Laplacian. *IEEE Trans. Autom. Control* **51**(1), 116–120 (2006)
24. Li, Z., Duan, Z., Chen, G.: On H_∞ and H_2 performance regions of multi-agent systems. *Automatica* **47**(4), 797–803 (2011)
25. Olfati-Saber, R.: Ultrafast consensus in small-world networks. In: Proceedings of the 2005 American Control Conference, pp. 2371–2378 (2005)
26. Xiao, L., Boyd, S.: Fast linear iterations for distributed averaging. *Syst. Control Lett.* **53**(1), 65–78 (2004)
27. Cortés, J.: Finite-time convergent gradient flows with applications to network consensus. *Automatica* **42**(11), 1993–2000 (2006)
28. Cao, Y., Ren, W., Chen, F., Zong, G.: Finite-time consensus of multi-agent networks with inherent nonlinear dynamics under an undirected interaction graph. In: Proceedings of the 2011 American Control Conference, pp. 4020–4025 (2011)

29. Wang, X., Hong, Y.: Finite-time consensus for multi-agent networks with second-order agent dynamics. In: Proceedings of the 17th World Congress, IFAC, pp. 15185–15190 (2008)
30. Cao, Y., Ren, W., Meng, Z.: Decentralized finite-time sliding mode estimators and their applications in decentralized finite-time formation tracking. *Syst. Control Lett.* **59**(9), 522–529 (2010)
31. Li, S., Du, H., Lin, X.: Finite-time consensus algorithm for multi-agent systems with double-integrator dynamics. *Automatica* **47**(8), 1706–1712 (2011)
32. Zhao, Y., Duan, Z., Wen, G., Zhang, Y.: Distributed finite-time tracking control for multi-agent systems: an observer-based approach. *Syst. Control Lett.* **62**(1), 22–28 (2013)
33. Hu, J., Hong, Y.: Leader-following coordination of multi-agent systems with coupling time delays. *Phys. A* **374**(2), 853–863 (2007)
34. Hong, Y., Xu, Y., Huang, J.: Finite-time control for robot manipulators. *Syst. Control Lett.* **46**(4), 243–253 (2002)
35. Chu, T., Wang, L., Chen, T., Mu, S.: Complex emergent dynamics of anisotropic swarms: convergence versus oscillation. *Chaos Solitons Fractals* **30**(4), 875–885 (2006)
36. Chen, G., Lewis, F., Xie, L.: Finite-time distributed consensus via binary control protocols. *Automatica* **47**(5), 1962–1968 (2011)
37. Osborn, G.: Mnemonic for hyperbolic formulae. *Math. Gaz.* **2**(34), 189 (1902)
38. Zhao, Y., Duan, Z., Wen, G.: Finite-time consensus for second-order multi-agent systems with saturated control protocols. *IET Control Theory Appl.* **9**(3), 312–319 (2014)
39. Filippov, A.: Differential Equations with Discontinuous Righthand Side. Kluwer, Norwell (1988)
40. Levant, A.: Principles of 2-sliding mode design. *Automatica* **43**(4), 576–586 (2007)
41. Davila, A., Moreno, J. A., Fridman, L.: Optimal Lyapunov function section for reaching time estimation of super twisting algorithm. In: Proceedings of the 48th IEEE Conference on Decision and Control, pp. 8405–8410 (2009)
42. Zhao, Y., Duan, Z.: Distributed finite-time containment control for multi-agent systems with multiple dynamic leaders. In: Proceedings of the 25th Chinese Control and Decision Conference, pp. 298–303 (2013)
43. Zhao, Y., Duan, Z., Wen, G., Chen, G.: Distributed finite-time tracking for a multi-agent system under a leader with bounded unknown acceleration. *Syst. Control Lett.* **81**(7), 8–13 (2015)
44. Zhao, Y., Duan, Z.: Finite-time containment control without velocity and acceleration measurements. *Nonlinear Dynamics* (2015) doi:[10.1007/s11071-015-2154-6](https://doi.org/10.1007/s11071-015-2154-6)
45. Wang, L., Xiao, F.: Finite-time consensus problems for networks of dynamic agents. *IEEE Trans. Autom. Control* **55**(4), 950–955 (2010)
46. Meng, Z., Ren, W., You, Z.: Distributed finite-time attitude containment control for multiple rigid bodies. *Automatica* **46**(12), 2092–2099 (2010)
47. Davila, J., Fridman, L., Levant, A.: Second-order sliding-mode observer for mechanical systems. *IEEE Trans. Autom. Control* **50**(11), 1785–1789 (2005)
48. Zhao, Y., Duan, Z., Wen, G.: Distributed finite-time tracking of multiple Euler-Lagrange dynamics without velocity measurements. *Int. J. Robust Nonlinear Control* **25**(11), 1688–1703 (2014)

Chapter 8

Composite Finite-Time Containment Control for Disturbed Second-Order Multi-agent Systems

Xiangyu Wang and Shihua Li

Abstract In this chapter, the distributed finite-time containment control problem is investigated for second-order multi-agent systems with external disturbances. By combining finite-time control and finite-time disturbance observer techniques together, a kind of feedforward-feedback composite distributed controllers are proposed. Under these distributed controllers, the effects of the disturbances on the system states are removed in a finite time and the followers globally converge to the convex hull spanned by the leaders in a finite time as well. Simulations illustrate the effectiveness of the proposed control algorithms.

8.1 Introduction

In recent years, distributed cooperative control of multi-agent systems has been a hot topic. The reasons mainly lie in two aspects. For one thing, it has broad applications, e.g., formation control of multiple mobile robots [1] or unmanned aerial vehicles [2], cooperative assembly of multiple manipulators [3], distributed fusion for sensor networks [4], etc. For another, it has several advantages than the traditional centralized coordination control approaches [5], e.g., better efficiency, higher robustness, less communication requirement, etc.

In the distributed cooperative control field, a most fundamental problem is the consensus problem (which is also called as the synchronization problem in complex networks [6, 7]). The consensus of multi-agent systems means that all the agents reach the agreement on a common state by implementing appropriate consensus

X. Wang · S. Li (✉)
School of Automation, Southeast University,
Nanjing 210096, Jiangsu, People's Republic of China
e-mail: lsh@seu.edu.cn

X. Wang
e-mail: w.x.y@seu.edu.cn

X. Wang · S. Li
Key Laboratory of Measurement and Control of Complex Systems of Engineering,
Ministry of Education, Nanjing 210096, People's Republic of China

controllers. Generally speaking, consensus problems can be classified into two categories, i.e., leaderless and leader-follower (with one leader) consensus. Recently, consensus algorithms have been extensively studied for first-order [8–10], second-order [11–14] and high-order [15–17] multi-agent systems.

As an extension of consensus, a more challenging problem is the containment control problem for multi-agent systems with multiple leaders. In this case, the control objective is to drive the states of the followers into the convex hull spanned by those of the leaders. The study of containment control stems from numerous natural phenomena and potential applications [18]. For example, the male silkworm moths always end up in the convex hull spanned by all the female silkworm moths by detecting pheromone released by females; for a vehicle group moving to a target place, the followers will stay in the safe area formed by the leaders when close to the hazardous obstacles, where the vehicles which are equipped necessary sensors to detect the obstacles play the role of the leaders and the others are the followers.

Due to the potential applications and importance of the containment control problem, several kinds of containment control algorithms have been proposed recently [19–23]. Note that all these control algorithms provide asymptotic convergence, which means that convergence rates of the closed-loop systems are at best exponential with an infinite convergence time. In other words, the states of the followers can not converge to the convex hull spanned by those of the leaders in a finite time. To this end, considering the convergence rates, finite-time containment control algorithms are more desired.

Besides faster convergence rates, the closed-loop systems with finite-time convergence usually demonstrate some other superiorities, such as better disturbance rejection properties and better robustness against uncertainties [13, 24, 25]. Because of the above superiorities, some kinds of finite-time containment control algorithms have been developed [26–29]. Although having aforementioned advantages, the finite-time feedback control methods proposed in these literature still belong to passive anti-disturbance control methods. In detail, they achieve the disturbance rejection goal through feedback regulation based on the tracking errors between the measured outputs and their setpoints. Hence, these controllers can not react directly and fast enough in the presence of strong disturbances, even though they can finally suppress the disturbances via feedback regulation in a relatively slow way. To this end, the disturbance rejection properties of the closed-loop systems under the finite-time feedback controllers still need to be enhanced.

Fortunately, disturbance observer based control (DOBC) [30] provides an effective way to enhance the disturbance rejection properties of the closed-loop systems under passive anti-disturbance control methods. Specifically, DOBC combines a disturbance observer (DO) and a conventional feedback controller together to form a feedforward-feedback composite controller. Since DOBC has faster responses, patch features and less conservativeness in handling the disturbances [30], DOBC methods have been widely used for control of individual systems [31–36] and multi-agent systems [37, 38]. But for multi-agent systems, the finite-time control results by using DOBC are still very limited.

This chapter aims to tackle the finite-time containment control problem for a class of disturbed second-order multi-agent systems. By integrating the finite-time observer and the adding a power integrator techniques together, a kind of feedforward-feedback composite distributed controllers are derived. Under these controllers, the effects of the disturbances on the system states are removed in a finite time and the followers move to the convex hull spanned by the leaders in a finite time as well. This chapter applies DOBC into containment control and achieves finite-time convergence for the closed-loop systems, which provides a novel way to solve the containment control problem of disturbed multi-agent systems.

The remainder of this chapter is organized as follows. In Sect. 8.2, some useful preliminaries and problem formulation are exhibited. In Sect. 8.3, the main result, i.e., the distributed finite-time containment control scheme, is presented. Some simulations are performed in Sect. 8.4. Finally, conclusions are drawn in Sect. 8.5.

8.2 Preliminaries and Problem Formulation

8.2.1 Notations

Denote $\text{sig}^\alpha(x) = \text{sgn}(x)|x|^\alpha$, where $x, \alpha \in \mathbb{R}$ and $\text{sgn}(\cdot)$ is the standard sign function. Given a vector $x = [x_1, \dots, x_n]^T \in \mathbb{R}^n$ and $\alpha \in \mathbb{R}$, let $[x]_i$ denote the i th element of x , $i = 1, \dots, n$. Moreover, denote $x^\alpha = [x_1^\alpha, \dots, x_n^\alpha]^T$, $\text{sig}^\alpha(x) = [\text{sig}^\alpha(x_1), \dots, \text{sig}^\alpha(x_n)]^T$, and $\text{sgn}(x) = [\text{sgn}(x_1), \dots, \text{sgn}(x_n)]^T$. Let $\|x\|_1 = \sum_{i=1}^n |x_i|$, $\|x\|_2 = \sqrt{x^T x}$, $\|x\|_\infty = \max_{i=1, \dots, n} |x_i|$ denote the 1-norm, Euclidean norm, and infinity norm of vector x , respectively. Let $P > 0$ denote a symmetric positive definite matrix P . Let $\lambda_{\max}(P)$ and $\lambda_{\min}(P)$ denote the maximum and minimum eigenvalues of matrix P , respectively. Denote $\mathbf{1}_n = [1, \dots, 1]^T \in \mathbb{R}^n$. Let I_p denote $p \times p$ identity matrix, where p is a positive integer.

8.2.2 Useful Lemmas and Definitions

Lemma 8.1 [24] Consider the system $\dot{x} = f(x)$, $f(0) = 0$, $x \in \mathbb{R}^n$, there exist a positive definite continuous function $V(x) : U \rightarrow \mathbb{R}$, real numbers $c > 0$ and $\alpha \in (0, 1)$, and an open neighbor $U_0 \subset U$ of the origin such that

$$\dot{V}(x) + c V^\alpha(x) \leq 0, \quad x \in U_0 \setminus \{0\}.$$

Then $V(x)$ approaches 0 in a finite time T . In addition, the finite convergence time T satisfies that $T \leq \frac{V^{1-\alpha}(x(0))}{c(1-\alpha)}$.

Lemma 8.2 [39] For any real numbers $x_i, i = 1, \dots, n$ and $0 < q \leq 1$, the following inequality holds

$$\left(\sum_{i=1}^n |x_i| \right)^q \leq \sum_{i=1}^n |x_i|^q.$$

If $0 < q = q_1/q_2 \leq 1$, where q_1, q_2 are odd integers, then

$$|x^q - y^q| \leq 2^{1-q} |x - y|^q.$$

Lemma 8.3 [39] If $c > 0, d > 0$ and $\gamma(x, y) > 0$ is a real-valued function for $x \in \mathbb{R}, y \in \mathbb{R}$, then

$$|x|^c |y|^d \leq \frac{c\gamma(x, y)|x|^{c+d}}{c+d} + \frac{d\gamma^{-c/d}(x, y)|y|^{c+d}}{c+d}.$$

Lemma 8.4 [40] Given matrices A and B with compatible sizes, then $(A \otimes B)^T = A^T \otimes B^T, (A \otimes I_p)(B \otimes I_p) = AB \otimes I_p$, where \otimes denotes the Kronecker product.

Definition 8.1 [26] Let X be a set in a real vector space $V \subseteq R^p$, where p is a positive integer. The convex hull $\mathbf{Co}(X)$ of the set X is defined as

$$\mathbf{Co}(X) = \left\{ \sum_{i=1}^k a_i x_i \mid x_i \in X, a_i \in \mathbb{R}, a_i \geq 0, \sum_{i=1}^k a_i = 1, k = 1, 2, \dots \right\}.$$

Consider the nonlinear system

$$\dot{x} = u + d(t), \quad (8.1)$$

where x is the state, u is the control input, and $d(t)$ is the external disturbance.

Lemma 8.5 [41] If $d(t)$ is m times differentiable and $d^{(m)}(t)$ has a known Lipschitz constant $L > 0$, and a nonlinear DO is designed as

$$\begin{cases} \dot{z}_0 = v_0 + u, \quad v_0 = -\lambda_0 L^{\frac{1}{m+1}} \text{sig}^{\frac{m}{m+1}}(z_0 - x) + z_1, \\ \dot{z}_l = v_l, \quad v_l = -\lambda_l L^{\frac{1}{m+1-l}} \text{sig}^{\frac{m-l}{m+1-l}}(z_l - v_{l-1}) + z_{l+1}, \\ \quad l = 1, \dots, m-1, \\ \dot{z}_m = -\lambda_m L \text{sgn}(z_m - v_{m-1}), \end{cases} \quad (8.2)$$

where $\lambda_0, \dots, \lambda_m > 0$ are large enough gains, and $z_0 = \hat{x}, z_l = \hat{d}^{(l-1)}, l = 1, \dots, m$ are the estimates of $x, d^{(l-1)}$, respectively. Then DO (8.2) is finite-time convergent.

8.2.3 Graph Theory Notions

Let $G = (\mathcal{V}, \mathcal{E}, \mathcal{A})$ be a directed graph, where $\mathcal{V} = \{1, \dots, n\}$ is the set of nodes, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges and $\mathcal{A} = [a_{ij}] \in \mathbb{R}^{n \times n}$ is the weighted adjacency matrix of the graph G . An edge (i, j) denotes that node j can access information from node i and i is said to be a neighbor of j , but not necessarily vice versa. The neighbor set of node i is denoted as $N_i = \{j \in \mathcal{V} | (j, i) \in \mathcal{E}\}$. In addition, an undirected graph G is defined such that $(j, i) \in \mathcal{E} \Leftrightarrow (i, j) \in \mathcal{E}$. In a directed graph, a directed path is a sequence of edges in the form $(k_1, k_2), (k_2, k_3), \dots, k_i \in \mathcal{V}$. An undirected path in an undirected graph is defined analogously. An undirected graph is connected if there is an undirected path between every pair of distinct nodes.

The adjacency matrix $\mathcal{A} = [a_{ij}] \in \mathbb{R}^{n \times n}$ associated with the directed graph G is defined such that $a_{ij} > 0$ if $(j, i) \in \mathcal{E}$ while $a_{ij} = 0$ otherwise. For an undirected graph, we assume that $a_{ij} = a_{ji}$. Moreover, we assume that $a_{ii} = 0, \forall i \in \mathcal{V}$. The Laplacian matrix $\mathcal{L} = [l_{ij}] \in \mathbb{R}^{n \times n}$ associated with \mathcal{A} is defined as $l_{ii} = \sum_{j \in N_i} a_{ij}$ and $l_{ij} = -a_{ij}$, where $i \neq j$. Obviously, zero is an eigenvalue of \mathcal{L} with an associated eigenvector $\mathbf{1}_n$. Note that matrix \mathcal{L} is symmetric for an undirected graph while not necessarily symmetric for a directed graph.

8.2.4 Problem Formulation

Firstly, in this chapter, the multi-agent systems to be studied are

$$\begin{aligned} \dot{x}_i(t) &= v_i(t), \quad \dot{v}_i(t) = u_i(t) + d_i(t), \quad i \in F, \\ \dot{x}_i(t) &= v_i(t), \quad \dot{v}_i(t) = u_i(t), \quad i \in L, \end{aligned} \tag{8.3}$$

where $x_i(t) = [x_{i,1}(t), \dots, x_{i,p}(t)]^T$, $v_i(t) = [v_{i,1}(t), \dots, v_{i,p}(t)]^T \in \mathbb{R}^p$ are respectively the position and velocity, and $u_i(t) = [u_{i,1}(t), \dots, u_{i,p}(t)]^T \in \mathbb{R}^p$ is the control input, and $d_i(t) = [d_{i,1}(t), \dots, d_{i,p}(t)]^T \in \mathbb{R}^p$ represents the external disturbance, associated with agent i , and p is a positive integer, and $F = \{1, \dots, n\}$, $L = \{n+1, \dots, n+m\}$ represent the follower set and the leader set, respectively.

Secondly, the communication topology of multi-agent system (8.3) is described by a directed graph $G_{n+m} = (\mathcal{V}_{n+m}, \mathcal{E}_{n+m}, \mathcal{A}_{n+m})$ with m leader nodes and n follower nodes. A node is called a follower if the node has at least one neighbor. Otherwise, the node is called a leader. $\mathcal{A}_{n+m} = [a_{ij}] \in \mathbb{R}^{(n+m) \times (n+m)}$ and $\mathcal{L}_{n+m} = [l_{ij}] \in \mathbb{R}^{(n+m) \times (n+m)}$ denote the adjacency and the Laplacian matrices of the graph G_{n+m} , respectively. For brevity, we use \mathcal{A} and \mathcal{L} to replace \mathcal{A}_{n+m} and \mathcal{L}_{n+m} later in this chapter, respectively. Let $G_n^F = (F, \mathcal{E}^F, A_F)$ and $G_m^L = (L, \mathcal{E}^L, A_L)$ denote the follower and the leader communication topologies, respectively. Assume that the leaders do not communicate with each other, which implies that $\mathcal{E}^L = \emptyset$. The communication between different followers are bidirectional, namely, G_n^F is an undirected graph. In addition, the communication between a leader and a follower

is unidirectional with the leader issuing the communication. Thus, each entry of the last m rows of the Laplacian matrix \mathcal{L}_{n+m} is zero and $\mathcal{L}_{n+m} = \begin{bmatrix} \mathcal{T} & \mathcal{T}_d \\ 0_{m \times n} & 0_{m \times m} \end{bmatrix}$, where $\mathcal{T} = [T_{ij}] \in \mathbb{R}^{n \times n}$, $\mathcal{T}_d \in \mathbb{R}^{n \times m}$. On the communication topology G_{n+m} of system (8.3), the following natural assumption is made.

Assumption 5 For each follower of multi-agent system (8.3), there exists at least one leader that has a path to the follower.

Lemma 8.6 [26] *Under Assumption 5, matrix \mathcal{T} is positive definite. In addition, each entry of $-\mathcal{T}^{-1}\mathcal{T}_d$ is nonnegative and each row sum of $-\mathcal{T}^{-1}\mathcal{T}_d$ is equal to one.*

For brevity, denote the vectors $x_F = [x_1^T, \dots, x_n^T]^T$, $v_F = [v_1^T, \dots, v_n^T]^T$, $u_F = [u_1^T, \dots, u_n^T]^T$ for the followers, denote the vectors $x_L = [x_{n+1}^T, \dots, x_{n+m}^T]^T$, $v_L = [v_{n+1}^T, \dots, v_{n+m}^T]^T$, $u_L = [u_{n+1}^T, \dots, u_{n+m}^T]^T$ for the leaders, and denote $x_d = [(x_1^d)^T, \dots, (x_n^d)^T]^T = -(\mathcal{T}^{-1}\mathcal{T}_d \otimes I_p)x_L$, $v_d = [(v_1^d)^T, \dots, (v_n^d)^T]^T = \dot{x}_d$, where $x_i^d = [x_{i,1}^d, \dots, x_{i,p}^d]^T$, $v_i^d = [v_{i,1}^d, \dots, v_{i,p}^d]^T \in \mathbb{R}^p$, $i \in F$. From Definition 8.1 and Lemma 8.6, $x_F \rightarrow x_d$ (i.e., $x_i \rightarrow x_i^d$, $i \in F$) means that x_i , $i \in F$ converge to the convex hull $\text{Co}\{x_j, j \in L\}$.

Based on the above descriptions, the objective of this chapter is to achieve distributed finite-time containment control for multi-agent system (8.3), i.e., to design distributed controllers for system (8.3) such that $x_i \rightarrow \text{Co}\{x_j, j \in L\}$, $i \in F$ (specifically, $x_i \rightarrow x_i^d$, $v_i \rightarrow v_i^d$, $i \in F$) in a finite time.

8.3 Main Result

Without loss of generality, the leaders of multi-agent system (8.3) are presumed to be dynamic. Actually, stationary leaders can also be regarded as dynamic ones but with zero velocities. Before designing the composite finite-time containment controllers, a distributed finite-time observer is proposed for the followers to obtain the accurate estimates of the weighted average of the leaders' velocities at first. Then, based on the velocity estimates, the composite distributed control scheme based on the combination of the adding a power integrator and finite-time observer techniques is proposed.

8.3.1 Distributed Finite-Time Observer Design

Before the control design, the following assumption is made.

Assumption 6 The control inputs of the leaders are unknown to any follower but their upper bound \bar{u}_0 (i.e., $|u_{i,k}(t)| \leq \bar{u}_0$, $\forall i \in L, k = 1, \dots, p, t \in [0, +\infty)$) is available to all the followers.

For multi-agent system (8.3), the distributed observer is designed as

$$\dot{\hat{v}}_i^d = -\rho_1 \text{sig}^\alpha \left(\sum_{j \in F \cup L} a_{ij} (\hat{v}_i^d - \hat{v}_j^d) \right) - \rho_2 \text{sgn} \left(\sum_{j \in F \cup L} a_{ij} (\hat{v}_i^d - \hat{v}_j^d) \right), \quad i \in F, \quad (8.4)$$

where $\hat{v}_i^d = [\hat{v}_{i,1}^d, \dots, \hat{v}_{i,p}^d]^T$, $i \in F$ denotes the estimate of v_i^d with respect to the i th follower and $\hat{v}_d = [\hat{v}_1^d, \dots, \hat{v}_n^d]^T$, and $\hat{v}_j^d = v_j$, $j \in L$, $\rho_1 > 0$, $\rho_2 > \bar{u}_0$, $0 < \alpha < 1$.

Proposition 8.1 *Under Assumptions 5 and 6, the distributed observer (8.4) is globally finite-time convergent, namely, $\hat{v}_i^d \rightarrow v_i^d$, $i \in F$ in a finite time T_0 .*

Proof Let $\tilde{v}_i^d = \hat{v}_i^d - v_i^d$, $i \in F$, $\tilde{v}_d = [(\tilde{v}_1^d)^T, \dots, (\tilde{v}_n^d)^T]^T$ denote the observation errors. Then the observation error dynamics can be written as

$$\tilde{v}_d = -\rho_1 \text{sig}^{\alpha_1} ((\mathcal{T} \otimes I_p) \tilde{v}_d) - \rho_2 \text{sgn} ((\mathcal{T} \otimes I_p) \tilde{v}_d) + (\mathcal{T}^{-1} \mathcal{T}_d \otimes I_p) u_L. \quad (8.5)$$

Choose the Lyapunov function as $V_o = \tilde{v}_d^T (\mathcal{T} \otimes I_p) \tilde{v}_d / 2$, which is positive definite and satisfies that $V_o \leq \lambda_{\max}(\mathcal{T}) \|\tilde{v}_d\|_2^2 / 2$. \dot{V}_o along (8.5) satisfies

$$\begin{aligned} \dot{V}_o &= \tilde{v}_d^T (\mathcal{T} \otimes I_p) [-\rho_1 \text{sig}^\alpha ((\mathcal{T} \otimes I_p) \tilde{v}_d) - \rho_2 \text{sgn} ((\mathcal{T} \otimes I_p) \tilde{v}_d) + (\mathcal{T}^{-1} \mathcal{T}_d \otimes I_p) u_L] \\ &\leq -\rho_1 \sum_{k=1}^{pn} |[(\mathcal{T} \otimes I_p) \tilde{v}_d]_k|^{1+\alpha} - \rho_2 \|(\mathcal{T} \otimes I_p) \tilde{v}_d\|_1 + \tilde{v}_d^T (\mathcal{T}_d \otimes I_p) u_L. \end{aligned} \quad (8.6)$$

By Lemma 8.2 and the fact that $0 < \alpha < 1$, it follows that

$$\sum_{k=1}^{pn} |[(\mathcal{T} \otimes I_p) \tilde{v}_d]_k|^{1+\alpha} \geq \|(\mathcal{T} \otimes I_p) \tilde{v}_d\|_2^{1+\alpha}. \quad (8.7)$$

Note that $\forall x = [x_1, \dots, x_n]^T$, $y = [y_1, \dots, y_n]^T \in \mathbb{R}^n$, the Hölder's inequality holds: $|x^T y| = |\sum_{k=1}^n x_k y_k| \leq (\sum_{k=1}^n |x_k|^{p_1})^{1/p_1} (\sum_{k=1}^n |y_k|^{q_1})^{1/q_1}$, where $\frac{1}{p_1} + \frac{1}{q_1} = 1$, $1 \leq p_1, q_1 \leq +\infty$. Then, by taking $x = \tilde{v}_d^T (\mathcal{T} \otimes I_p)$, $y = (\mathcal{T}^{-1} \mathcal{T}_d \otimes I_p) u_L$, $p_1 = 1$, $q_1 = +\infty$ in the above inequality, it holds that

$$\begin{aligned} \tilde{v}_d^T (\mathcal{T} \otimes I_p) (\mathcal{T}^{-1} \mathcal{T}_d \otimes I_p) u_L &\leq \|(\mathcal{T} \otimes I_p) \tilde{v}_d\|_1 \|(\mathcal{T}^{-1} \mathcal{T}_d \otimes I_p) u_L\|_\infty \\ &\leq \bar{u}_0 \|(\mathcal{T} \otimes I_p) \tilde{v}_d\|_1. \end{aligned} \quad (8.8)$$

Substituting (8.7) and (8.8) into (8.6) yields

$$\begin{aligned}\dot{V}_o &\leq -\rho_1 \|(\mathcal{T} \otimes I_p) \tilde{v}_d\|_2^{1+\alpha} - (\rho_2 - \bar{u}_0) \|(\mathcal{T}_d \otimes I_p) \tilde{v}_d\|_1, \\ &\leq -\rho_1 \|(\mathcal{T} \otimes I_p) \tilde{v}_d\|_2^{1+\alpha} - (\rho_2 - \bar{u}_0) \|(\mathcal{T}_d \otimes I_p) \tilde{v}_d\|_2.\end{aligned}\quad (8.9)$$

Since $\|(\mathcal{T} \otimes I_p) \tilde{v}_d\|_2 = (\tilde{v}_d^T (\mathcal{T} \otimes I_p)^2 \tilde{v}_d)^{1/2} \geq \lambda_{\min}(\mathcal{T}) \|\tilde{v}_d\|_2 \geq \frac{2^{1/2} \lambda_{\min}^{1/2}(\mathcal{T}) V_o^{1/2}}{\lambda_{\max}^{1/2}(\mathcal{T})}$. It follows from (8.9) that

$$\dot{V}_o \leq -\frac{2^{(1+\alpha)/2} \rho_1 \lambda_{\min}^{1+\alpha}(\mathcal{T})}{\lambda_{\max}^{(1+\alpha)/2}(\mathcal{T})} V_o^{(1+\alpha)/2} - \frac{2^{1/2} (\rho_2 - \bar{u}_0) \lambda_{\min}(\mathcal{T})}{\lambda_{\max}^{1/2}(\mathcal{T})} V_o^{1/2}. \quad (8.10)$$

With the given condition $\alpha \in (0, 1)$, $\rho_1 > 0$, $\rho_2 > \bar{u}_0$ and Lemma 8.1, it can be verified that $\tilde{v}_d(t) \rightarrow 0$ in a finite time T_0 and T_0 satisfies that

$$T_0 \leq \min \left\{ \frac{2^{(1-\alpha)/2} \lambda_{\max}^{(1+\alpha)/2}(\mathcal{T}) V_o^{(1-\alpha)/2}(0)}{(1-\alpha) \rho_1 \lambda_{\min}^{1+\alpha}(\mathcal{T})}, \frac{2^{1/2} \lambda_{\max}^{1/2}(\mathcal{T}) V_o^{1/2}(0)}{(\rho_2 - \bar{u}_0) \lambda_{\min}(\mathcal{T})} \right\}.$$

This completes the proof. \square

Remark 8.1 Actually, even though the upper bound \bar{u}_0 is unknown to some followers, ρ_2 can be set relatively large to guarantee the finite-time convergence of the distributed observer (8.4).

Remark 8.2 If $\rho_1 = 0$ or $\alpha = 0$ in the distributed observer (8.6), observer (8.6) reduces to the sliding-mode distributed observer proposed in [26]. It is not difficult to verify that observer (8.6) has a faster convergence rate than the sliding-mode observer in [26], due to the presence of the term $-\rho_1 \text{sig}^\alpha \left(\sum_{j \in F \cup L} a_{ij} (\hat{v}_i^d - \hat{v}_j^d) \right)$.

8.3.2 Composite Distributed Finite-Time Containment Controller Design

Assumption 7 The disturbance $d_{i,k}$, $i \in F$, $k = 1, \dots, p$ in system (8.3) is differentiable and $\dot{d}_{i,k}$ has a Lipschitz constant $L_{i,k}$.

Remark 8.3 Assumption 7 is borrowed from [41]. Based on this assumption, the result of Lemma 8.5 can be utilized to design disturbance observers for the followers. In fact, many kinds of disturbances satisfy Assumption 7, e.g., constant disturbances, ramp disturbances, and sinusoidal disturbances, etc.

Proposition 8.2 For multi-agent system (8.3), if Assumptions 5–7 hold and the distributed controller u_i , $i \in F$ is designed as

$$u_i = \dot{\hat{v}}_i^d - k_2 \left[(v_i - \hat{v}_i^d)^{1/q} + k_1^{1/q} \sum_{j \in F \cup L} a_{ij}(x_i - x_j) \right]^{2q-1} - \hat{d}_i, \quad i \in F, \quad (8.11)$$

where \hat{v}_i^d is the estimate of v_i^d generated from observer (8.4), and the control parameters satisfy

$$\begin{aligned} k_1 &\geq \frac{2^{1-q}}{1+q} + \frac{(\beta + n\eta)q}{1+q} + k_3, \quad k_3 > 0, \\ k_2 &\geq (2-q)2^{1-q}k_1^{1+1/q} \left[\frac{(k_1 + n\eta)2^{1-q}q}{k_1(1+q)} + \frac{\sigma}{k_1} + k_3 \right], \\ \beta &= \max_{\forall i \in F} \left\{ \sum_{j \in F \cup L} a_{ij} \right\}, \quad \eta = \max_{\forall i, j \in F} \{a_{ij}\}, \quad \sigma = \frac{(\beta + n\eta)(k_1 + 2^{1-q}) + \beta q 2^{1-q}}{1+q}, \end{aligned}$$

$1/2 < q = q_1/q_2 < 1$ with positive odd integers q_1, q_2 , and \hat{d}_i is the estimate of disturbance d_i generated from the following DO

$$\begin{cases} \dot{z}_{i,k}^0 = \phi_{i,k} + u_{i,k}, \quad \phi_{i,k} = -\lambda_{i,k}^0 L_{i,k}^{1/2} \text{sig}^{1/2}(z_{i,k}^0 - v_{i,k}) + z_{i,k}^1, & i \in F, \\ \dot{z}_{i,k}^1 = -\lambda_{i,k}^1 L_{i,k} \text{sgn}(z_{i,k}^1 - \phi_{i,k}), \end{cases} \quad (8.12)$$

where $\lambda_{i,k}^0, \lambda_{i,k}^1 \in \mathbb{R}$, $i \in F, k = 1, \dots, p$ are large enough gains and $z_{i,k}^0 = \hat{v}_{i,k}$, $\hat{v}_i = [\hat{v}_{i,1}, \dots, \hat{v}_{i,p}]^T$, $z_{i,k}^1 = \hat{d}_{i,k}$, $\hat{d}_i = [\hat{d}_{i,1}, \dots, \hat{d}_{i,p}]^T$, then $x_i(t), v_i(t), i \in F$ are bounded $\forall t \in [0, +\infty)$.

Proof See Appendix.

With the help of Proposition 8.2, the main result of this chapter can be stated as the following theorem.

Theorem 8.1 For multi-agent system (8.3) with multiple dynamic leaders, if Assumptions 5–7 hold and the distributed controller $u_i, i \in F$ is designed as (8.11), then $x_i \rightarrow \mathbf{Co}\{x_j, j \in L\}$ in a finite time, more specifically, $x_i \rightarrow x_i^d, v_i \rightarrow v_i^d, i \in F$ in a finite time.

Proof Denote the estimation errors of DO (8.12) as $\tilde{v}_i(t) = [\tilde{v}_{i,1}, \dots, \tilde{v}_{i,p}]^T = \hat{v}_i(t) - v_i(t)$, $\tilde{d}_i(t) = [\tilde{d}_{i,1}, \dots, \tilde{d}_{i,p}]^T = \hat{d}_i(t) - d_i(t)$. From the proof of Proposition 8.2, DO (8.12) is finite-time convergent, i.e., there is a time instant T_1 such that $\tilde{v}_i(t) = \tilde{d}_i(t) = 0, \forall i \in F, t \in [T_1, +\infty)$. Denote $T = \max\{T_0, T_1\}$. For the case of $0 < t < T$, it follows from Proposition 8.2 that $x_i(t), v_i(t), i \in F$ are bounded. Next, we focus on the global finite-time convergence proof for the case of $t \geq T$.

When $t \geq T$, it follows that $\hat{v}_i^d = v_i^d, i \in F$. Denote the tracking errors as $\bar{x}_i = [\bar{x}_{i,1}, \dots, \bar{x}_{i,p}]^T = x_i - x_i^d, \bar{v}_i = [\bar{v}_{i,1}, \dots, \bar{v}_{i,p}]^T = \dot{\bar{x}}_i, \bar{u}_i = [\bar{u}_{i,1}, \dots, \bar{u}_{i,p}]^T = u_i + d_i - \hat{v}_i^d, i \in F$, and $\bar{x}_j = 0, j \in L, \bar{x}_F = [\bar{x}_1^T, \dots, \bar{x}_n^T]^T, \bar{v}_F = [\bar{v}_1^T, \dots, \bar{v}_n^T]^T$,

$\bar{x}_L = [\bar{x}_{n+1}^T, \dots, \bar{x}_{n+m}^T]^T$. By applying the new notations to system (8.3), the tracking error dynamics of the followers and the leaders can be respectively written as

$$\dot{\bar{x}}_i(t) = \bar{v}_i(t), \quad \dot{\bar{v}}_i(t) = \bar{u}_i(t), \quad i \in F, \quad (8.13)$$

$$\bar{x}_j(t) = 0, \quad \dot{\bar{x}}_j(t) = 0, \quad \ddot{\bar{x}}_j(t) = 0, \quad j \in L. \quad (8.14)$$

The following proof is based on the adding a power integrator technique [39, 42] and it is composed of two steps. First, a virtual velocity \bar{v}_i^* is designed for each follower. Second, the distributed controller is designed for each follower such that $\bar{v}_i \rightarrow \bar{v}_i^*$ in a finite time and then global finite-time convergence of the closed-loop system (8.11)–(8.13) is guaranteed.

Step 1. (Virtual velocity design) Choose the following Lyapunov function

$$\begin{aligned} V_0 &= \frac{1}{2} \bar{x}_F^T (\mathcal{T} \otimes I_p) \bar{x}_F \\ &= \frac{1}{4} \sum_{i=1}^n \sum_{j=1}^n a_{ij} \|\bar{x}_i - \bar{x}_j\|_2^2 + \frac{1}{2} \sum_{i=1}^n \sum_{j=n+1}^{n+m} a_{ij} \|\bar{x}_i\|_2^2 \end{aligned} \quad (8.15)$$

By Assumption 5, V_0 is positive definite and differentiable. In addition, $V_0 \leq \frac{1}{2} \lambda_{\max}(\mathcal{T}) \bar{x}_F^T \bar{x}_F$, where $\lambda_{\max}(\mathcal{T}) > 0$ since $\mathcal{T} > 0$ (by Lemma 8.6). The derivative of V_0 along system (8.13) is

$$\dot{V}_0 = \bar{x}_F^T (\mathcal{T} \otimes I_p) \dot{\bar{x}}_F = \sum_{i=1}^n \left[\sum_{j=1}^{n+m} a_{ij} (\bar{x}_i - \bar{x}_j)^T \right] \bar{v}_i. \quad (8.16)$$

Denote $w_i = [w_{i,1}, \dots, w_{i,p}]^T = \sum_{j=1}^{n+m} a_{ij} (\bar{x}_i - \bar{x}_j)$, $i \in F$. Then $[w_1^T, \dots, w_n^T]^T = (\mathcal{T} \otimes I_p) \bar{x}_F$ and $\sum_{i=1}^n w_i^T w_i = \bar{x}_F^T (\mathcal{T} \otimes I_p)^T (\mathcal{T} \otimes I_p) \bar{x}_F = \bar{x}_F^T (\mathcal{T}^2 \otimes I_p) \bar{x}_F$. Thus,

$$\sum_{i=1}^n w_i^T w_i \geq \lambda_{\min}(\mathcal{T}^2) \bar{x}_F^T \bar{x}_F \geq \frac{2\lambda_{\min}(\mathcal{T}^2) V_0}{\lambda_{\max}(\mathcal{T})}, \quad (8.17)$$

where $\lambda_{\min}(\mathcal{T}^2) > 0$ since $\mathcal{T}^2 > 0$. Take the virtual velocity as

$$\bar{v}_i^* = [\bar{v}_{i,1}^*, \dots, \bar{v}_{i,p}^*]^T = -k_1 w_i^q, \quad i \in F, \quad (8.18)$$

where $k_1 > 0$ to be determined, and $1/2 < q = q_1/q_2 < 1$ with positive odd integers q_1, q_2 . With the help of (8.18), (8.16) becomes

$$\dot{V}_0 = -k_1 \sum_{i=1}^n w_i^T w_i^q + \sum_{i=1}^n w_i^T (\bar{v}_i - \bar{v}_i^*). \quad (8.19)$$

Step 2. (Distributed controller design) Denote $\xi_i = [\xi_{i,1}, \dots, \xi_{i,p}]^T = \bar{v}_i^{1/q} - \bar{v}_i^{*1/q}$, $i \in F$ and $r = 1 + q$. Choose the following Lyapunov function

$$V = V_0 + \sum_{i=1}^n \sum_{k=1}^p V_{i,k}, \quad (8.20)$$

where $V_{i,k} = \frac{1}{(2-q)2^{1-q}k_1^{1+1/q}} \int_{\bar{v}_{i,k}^*}^{\bar{v}_{i,k}} (s^{1/q} - \bar{v}_{i,k}^{*1/q})^{2-q} ds$, $\bar{v}_{i,k}^* = -k_1 w_{i,k}^q$, $i \in F$, $k = 1, \dots, p$. From Propositions B1 and B2 in [42], $V_{i,k}$ (also V) is differentiable, positive definite and proper $\forall i \in F$, $k = 1, \dots, p$. Moreover, based on the fact $0 < q < 1$ and Lemma 8.2, it can be obtained that

$$\begin{aligned} V_{i,k} &\leq \frac{1}{(2-q)2^{1-q}k_1^{1+1/q}} |\bar{v}_{i,k} - \bar{v}_{i,k}^*| |\xi_{i,k}|^{2-q} \\ &\leq \frac{1}{(2-q)k_1^{1+1/q}} \xi_{i,k}^2, \quad i \in F, k = 1, \dots, p. \end{aligned} \quad (8.21)$$

Then by (8.17) and (8.21), there is $c = \max \left\{ \frac{\lambda_{\max}(\mathcal{T})}{2\lambda_{\min}(\mathcal{T}^2)}, \frac{1}{(2-q)k_1^{1+1/q}} \right\}$ such that

$$V = V_0 + \sum_{i=1}^n \sum_{k=1}^p V_{i,k} \leq c \sum_{i=1}^n \sum_{k=1}^p (w_{i,k}^2 + \xi_{i,k}^2), \quad (8.22)$$

Next, we estimate the terms in $\dot{V} = \dot{V}_0 + \sum_{i=1}^n \sum_{k=1}^p \dot{V}_{i,k}$ from left to right. First, by Lemmas 8.2–8.3, it follows from (8.19) that

$$\begin{aligned} \dot{V}_0 &\leq -k_1 \sum_{i=1}^n \sum_{k=1}^p w_{i,k}^r + 2^{1-q} \sum_{i=1}^n \sum_{k=1}^p |\phi_{i,k}| |\xi_{i,k}|^q \\ &\leq -k_1 \sum_{i=1}^n \sum_{k=1}^p w_{i,k}^r + 2^{1-q} \sum_{i=1}^n \sum_{k=1}^p \left(\frac{w_{i,k}^r}{r} + \frac{q\xi_{i,k}^r}{r} \right) \end{aligned} \quad (8.23)$$

Second, $\dot{V}_{i,k}$ along system (8.13) satisfies ($i \in F$, $k = 1, \dots, p$)

$$\dot{V}_{i,k} = -\frac{1}{2^{1-q}k_1^{1+1/q}} \frac{d\bar{v}_{i,k}^{*1/q}}{dt} \int_{\bar{v}_{i,k}^*}^{\bar{v}_{i,k}} (s^q - \bar{v}_{i,k}^{*1/q})^{1-q} ds + \frac{\xi_{i,k}^{2-q} \bar{u}_{i,k}}{(2-q)2^{1-q}k_1^{1+1/q}}. \quad (8.24)$$

From (8.18), it follows that $d\bar{v}_{i,k}^{*1/q}/dt = -k_1^{1/q} \sum_{j=1}^{n+m} a_{ij} (\bar{v}_{i,k} - \bar{v}_{j,k}) \leq k_1^{1/q} (\beta |\bar{v}_{i,k}| + \eta \sum_{j=1}^n |\bar{v}_{j,k}|)$, $i \in F$, where $\beta = \max_{\forall i \in F} \left\{ \sum_{j \in F \cup L} a_{ij} \right\}$ and $\eta = \max_{\forall i,j \in F} \{a_{ij}\}$.

In addition, by Lemma 8.2, $\int_{\bar{V}_{i,k}^*}^{\bar{v}_{i,k}} (s^{1/q} - \bar{v}_{i,k}^{*1/q})^{1-q} ds \leq |\bar{v}_{i,k} - \bar{v}_{i,k}^*| |\xi_{i,k}|^{1-q} \leq 2^{1-q} |\xi_{i,k}|$. Based on the above analysis, it follows from (8.24) that

$$\dot{V}_{i,k} \leq \frac{1}{k_1} \left(\beta |\bar{v}_{i,k}| + \eta \sum_{j=1}^n |\bar{v}_{j,k}| \right) |\xi_{i,k}| + \frac{\xi_{i,k}^{2-q} \bar{u}_{i,k}}{(2-q)2^{1-q} k_1^{1+1/q}}, \quad i \in F, k = 1, \dots, p. \quad (8.25)$$

From (8.18) and Lemma 8.2, it holds that $|\bar{v}_{j,k}| \leq |\bar{v}_{j,k}^*| + |\bar{v}_{j,k} - \bar{v}_{j,k}^*| \leq k_1 |w_{j,k}|^q + 2^{1-q} |\xi_{j,k}|^q$. By Lemma 8.3, it holds that $|\bar{v}_{j,k}| |\xi_{i,k}| \leq (k_1 |w_{j,k}|^q + 2^{1-q} |\xi_{j,k}|^q) |\xi_{i,k}| \leq \frac{k_1 q}{r} w_{j,k}^r + \frac{2^{1-q} q}{r} \xi_{j,k}^r + \frac{k_1 + 2^{1-q}}{r} \xi_{i,k}^r$. Then, by applying the above inequalities into (8.25) yields ($i \in F, k = 1, \dots, p$)

$$\dot{V}_{i,k} \leq \frac{\beta q}{r} w_{i,k}^r + \frac{\sigma}{k_1} \xi_{i,k}^r + \frac{\eta q}{r} \sum_{j=1}^n w_{j,k}^r + \frac{\eta 2^{1-q} q}{k_1 r} \sum_{j=1}^n \xi_{j,k}^r + \frac{\xi_{i,k}^{2-q} \bar{u}_{i,k}}{(2-q)2^{1-q} k_1^{1+1/q}}, \quad (8.26)$$

where $\sigma = \frac{(\beta+n\eta)(k_1+2^{1-q})+\beta q 2^{1-q}}{r}$. Putting (8.20), (8.23), and (8.26) together yields

$$\begin{aligned} \dot{V} &\leq - \left[k_1 - \frac{2^{1-q}}{r} - \frac{(\beta+n\eta)q}{r} \right] \sum_{i=1}^n \sum_{k=1}^p w_{i,k}^r \\ &\quad + \left[\frac{(k_1+n\eta)2^{1-q}q}{k_1 r} + \frac{\sigma}{k_1} \right] \sum_{i=1}^n \sum_{k=1}^p \xi_{i,k}^r + \frac{1}{(2-q)2^{1-q} k_1^{1+1/q}} \sum_{i=1}^n \sum_{k=1}^p \xi_{i,k}^{2-q} \bar{u}_{i,k}. \end{aligned} \quad (8.27)$$

Denote $\mathcal{B} = [b_{ij}] = -\mathcal{T}^{-1} \mathcal{I}_d$. For one thing, $x_d = (\mathcal{B} \otimes I_p) x_L$ and then $\sum_{j \in F} T_{ij} x_j^d = \sum_{j=1}^n T_{ij} \sum_{k=1}^m b_{jk} x_{k+n} = \sum_{k=1}^m \left(\sum_{j=1}^n T_{ij} b_{jk} \right) x_{k+n}$. For another thing, the fact that $\mathcal{T}(-\mathcal{T}^{-1} \mathcal{I}_d) = -\mathcal{I}_d$ indicates that $\sum_{j=1}^n T_{ij} b_{jk} = -l_{ik+n}, i \in F, k = 1, \dots, m$, where l_{ik+n} is equal to the (i, k) entry of matrix \mathcal{I}_d . Therefore, $\sum_{j \in F} T_{ij} x_j^d = -\sum_{j \in L} l_{ij} x_j, i \in F$. Then it follows that $\sum_{j \in F \cup L} a_{ij} (\bar{x}_i - \bar{x}_j) = \sum_{j \in F} T_{ij} \bar{x}_j = \sum_{j \in F \cup L} a_{ij} (x_i - x_j)$. If u_i is taken as (8.11), by noting that $\bar{u}_i = u_i + d_i - \dot{v}_i^d, i \in F$, \bar{u}_i can also be described as

$$\bar{u}_{i,k} = -k_2 \xi_{i,k}^{2q-1}, \quad i \in F, k = 1, \dots, p, \quad (8.28)$$

where $k_1 \geq \frac{2^{1-q}}{r} + \frac{(\beta+n\eta)q}{r} + k_3$, $k_2 \geq (2-q)2^{1-q} k_1^{1+1/q} \left[\frac{(k_1+n\eta)2^{1-q}q}{k_1 r} + \frac{\sigma}{k_1} + k_3 \right]$ and $k_3 > 0$. Substituting (8.28) into (8.27) yields

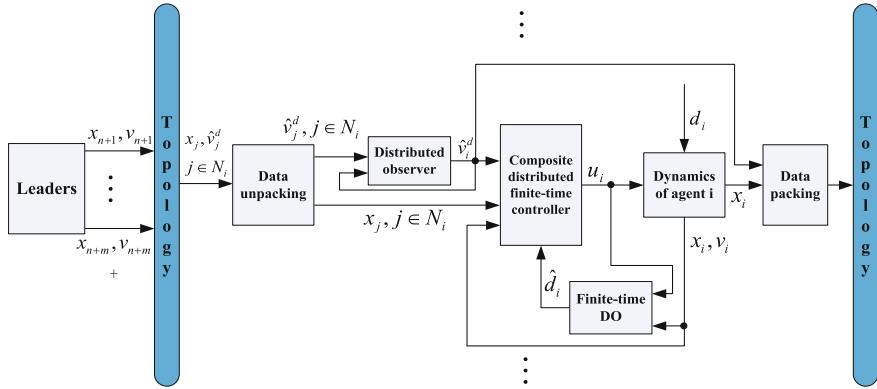


Fig. 8.1 The block diagram of the closed-loop system (8.3), (8.4), (8.11) and (8.12)

$$\dot{V} \leq -k_3 \sum_{i=1}^n \sum_{k=1}^p (w_{i,k}^r + \xi_{i,k}^r). \quad (8.29)$$

Since $0 < r/2 < 1$, by Lemma 8.2, it follows from (8.22) and (8.29) that

$$\dot{V} + \frac{k_3}{c^{r/2}} V^{r/2} \leq 0, \quad (8.30)$$

which means that V reaches zero in a finite time (by Lemma 8.1). Then $\bar{x}_F \rightarrow 0$, $\bar{v}_F \rightarrow 0$, i.e., $x_i \rightarrow x_i^d$ and $v_i \rightarrow v_i^d$, $i \in F$ in a finite time with controller (8.11). This completes the proof. \square

The block diagram of the closed-loop system (8.3), (8.4), (8.11) and (8.12) is shown in Fig. 8.1.

Remark 8.4 The composite distributed controller (8.11) is composed of two parts, i.e., the feedback control part and the feedforward control part. The feedback terms $\dot{\hat{v}}_i^d - k_2 \left[(v_i - \hat{v}_i^d)^{1/q} + k_1^{1/q} \sum_{j \in F \cup L} a_{ij} (x_i - x_j) \right]^{2q-1}$ are used to make the i th ($i \in F$) follower converge to the leaders' convex hull $\text{Co}\{x_j, j \in L\}$ in a finite time, and these terms are obtained through a recursive design process based on the adding a power integrator technique [39, 42]. The feedforward term \hat{d}_i is generated from the DO (8.12) and it is used to compensate the disturbance d_i such that the global finite-time convergence of the whole closed-loop system can be achieved. In the absence of disturbances, it can be derived from (8.12) that

$$\begin{cases} \dot{\hat{v}}_{i,k} = -\lambda_{i,k}^0 L_{i,k}^{1/2} \text{sig}^{1/2}(\hat{v}_{i,k}) + \hat{d}_{i,k}, \\ \dot{\hat{d}}_{i,k} = -\lambda_{i,k}^1 L_{i,k} \text{sgn}(\hat{d}_{i,k} - \dot{\hat{v}}_{i,k}). \end{cases} \quad (8.31)$$

Then $\tilde{d}_i(t) = 0, \forall i \in F, t \in [0, +\infty)$ if the initial states of the estimates are selected as $\hat{v}_{i,k}(0) = z_{i,k}^0(0) = v_{i,k}(0), \hat{d}_{i,k}(0) = z_{i,k}^1(0) = 0, \forall i \in F, k = 1, \dots, p$. In this case, the composite distributed controller (8.11) degrades to the following nonsmooth controller

$$u_i = \dot{\hat{v}}_i^d - k_2 \left[(v_i - \hat{v}_i^d)^{1/q} + k_1^{1/q} \sum_{j \in F \cup L} a_{ij}(x_i - x_j) \right]^{2q-1}, \quad i \in F, \quad (8.32)$$

where the parameters satisfy the same sufficient conditions as controller (8.11). For one thing, this indicates that the proposed composite distributed control scheme does not sacrifice the nominal control performance. For another thing, in the presence of disturbances, due to the continuity, controller (8.32) can not remove the effects of the disturbances completely from the output channels even in the steady states. Therefore, the composite distributed controller (8.11) provides a better disturbance rejection property and retains the nominal control performance. This will be verified via simulations in the next section.

Actually, the result of Theorem 8.1 also covers the case of multiple stationary leaders. The only difference is that for multi-agent system (8.3) with multiple stationary leaders (i.e., the leaders' velocities are all zeros), the distributed observer is not needed anymore. More specifically, without further proof, the following corollary can be given.

Corollary 8.1 *For multi-agent system (8.3) with multiple stationary leaders, if Assumptions 5 and 7 hold and the distributed controller $u_i, i \in F$ is designed as*

$$u_i = -k_2 \left[v_i^{1/q} + k_1^{1/q} \sum_{j \in F \cup L} a_{ij}(x_i - x_j) \right]^{2q-1} - \hat{d}_i, \quad i \in F, \quad (8.33)$$

where \hat{d}_i also comes from DO (8.12) and the control parameters satisfy the same sufficient conditions as controller (8.11), then $x_i \rightarrow \text{Co}\{x_j, j \in L\}$ in a finite time, more specifically, $x_i \rightarrow x_i^d, v_i \rightarrow 0, i \in F$ in a finite time.

8.4 Numerical Simulations

In this section, some simulations are conducted to illustrate the effectiveness of the control scheme proposed in Theorem 8.1. A group of agents with 4 leaders and 4 followers in the two-dimensional space are considered, i.e., $m = 4, n = 4$. Comparisons will be made between the performances of the two closed-loop systems under the composite distributed controller (8.11) and the reduced distributed controller (8.32) without disturbance compensation.

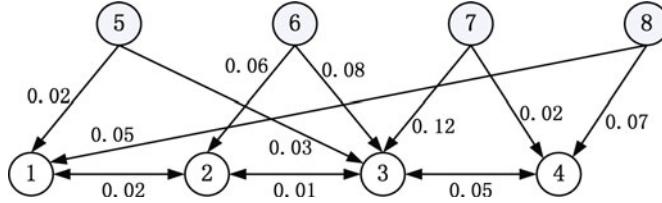


Fig. 8.2 The communication topology among the agents with $F = \{1, 2, 3, 4\}$, $L = \{5, 6, 7, 8\}$

The communication topology among the agents is shown in Fig. 8.2 with $F = \{1, 2, 3, 4\}$ and $L = \{5, 6, 7, 8\}$. The external disturbances are assumed to be $d_{1,1}(t) = 0.2t - 3$, $d_{1,2}(t) = 5 \sin(3t) + 5$ (follower 1), $d_{2,1}(t) = 3$, $d_{2,2}(t) = 0.1t + 1$ (follower 2), $d_{3,1}(t) = 2$, $d_{3,2}(t) = -3$ (follower 3), $d_{4,1}(t) = 2 \cos(5t) + 6$, $d_{4,2}(t) = -3 \sin(10t) - 2$ (follower 4). The Lipschitz constants for \dot{d}_i are chosen as $L_{1,1} = 1$, $L_{1,2} = 30$ (follower 1), $L_{2,1} = L_{2,2} = 1$ (follower 2), $L_{3,1} = 1$, $L_{3,2} = 2$ (follower 3), $L_{4,1} = 20$, $L_{4,2} = 50$ (follower 4).

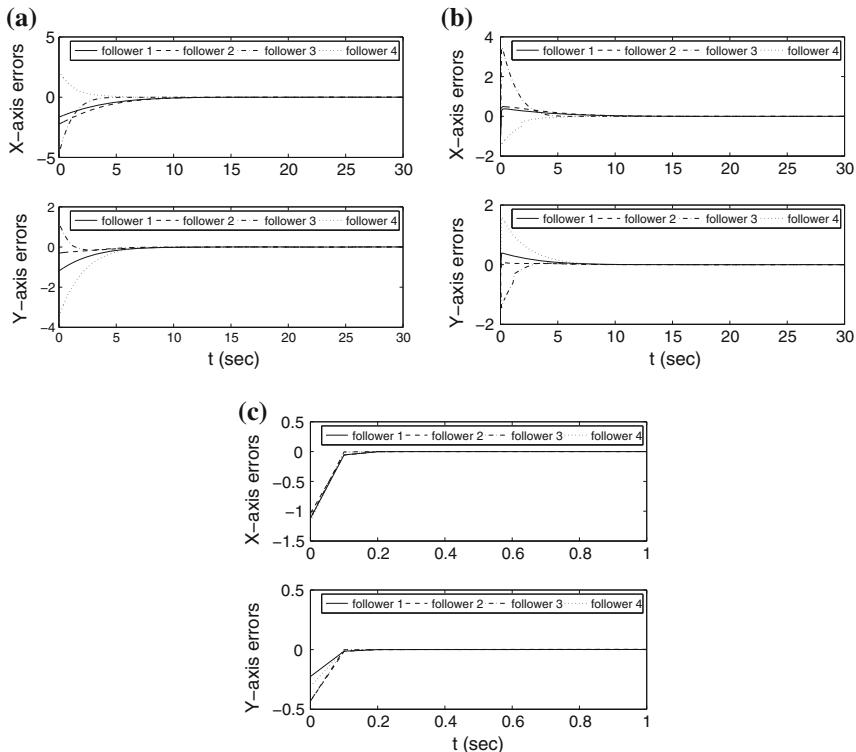


Fig. 8.3 Response curves under the composite distributed controller (8.11): **a** Position tracking errors $x_i - x_i^d$, $i \in F$. **b** Velocity tracking errors $v_i - v_i^d$, $i \in F$. **c** Observation errors $\hat{v}_i^d - v_i^d$, $i \in F$ of the distributed observer (8.4)

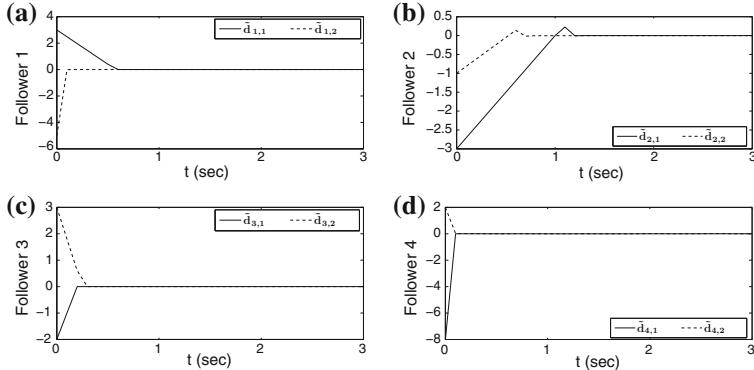


Fig. 8.4 Disturbance estimation errors from DO (8.12): **a** Disturbance estimation errors of follower 1. **b** Disturbance estimation errors of follower 2. **c** Disturbance estimation errors of follower 3. **d** Disturbance estimation errors of follower 4

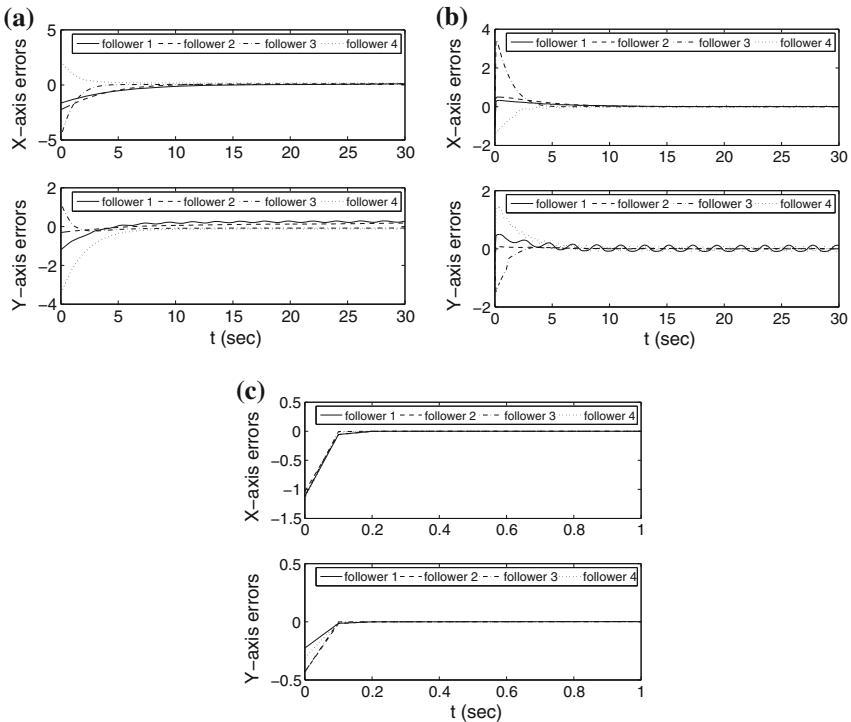


Fig. 8.5 Response curves under the reduced distributed controller (8.32): **a** Position tracking errors $x_i - x_i^d, i \in F$. **b** Velocity tracking errors $v_i - v_i^d, i \in F$. **c** Observation errors $\hat{v}_i^d - v_i^d, i \in F$ of the distributed observer (8.4)

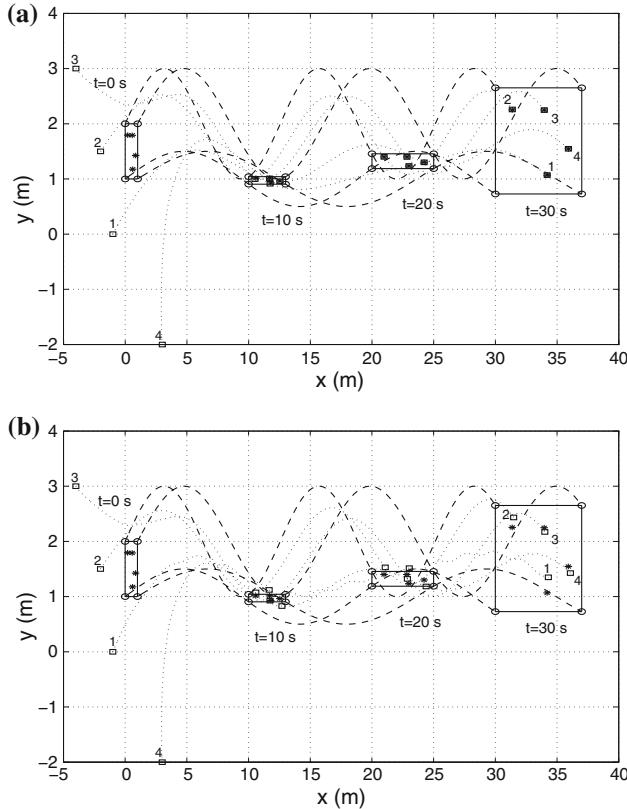


Fig. 8.6 Trajectories of agents 1 to 8, where the circles denote the leaders, the squares denote the followers, and the asterisks denote the desired positions of the followers: **a** Trajectories under the composite distributed controller (8.11). **b** Trajectories under the reduced distributed controller (8.32)

The motion equations of the leaders are $x_5(t) = [t, 0.5 \sin(t/3) + 1]^T$, $x_6(t) = [t, \sin(t/2) + 2]^T$, $x_7(t) = [1.2t + 1, \sin(t/2) + 2]^T$, $x_8(t) = [1.2t + 1, 0.5 \sin(t/3) + 1]^T$, $\forall t \geq 0$. The followers 1–4 are assumed to be static at $t = 0$ and their initial coordinates are $x_1(0) = [-1, 0]^T$, $x_2(0) = [-2, 1.5]^T$, $x_3(0) = [-4, 3]^T$, $x_4(0) = [3, -2]^T$. The initial states of the distributed observer (8.4) are $\hat{v}_i^d(0) = [0, 0]^T$, $i \in F$. The initial states of the DO (8.12) are $\hat{v}_i(0) = \hat{d}_i(0) = [0, 0]^T$, $i \in F$.

According to the sufficient conditions given in Propositions 8.1 and 8.2, parameters of the distributed observer (8.4), the composite distributed controller (8.11), and the DO (8.12) are chosen as $q = 11/13$, $k_1 = 2.8272$, $k_2 = 36.7896$, $\rho_1 = 20$, $\rho_2 = 20$, $\alpha = 0.8$, $\lambda_{1,1}^0 = 10$, $\lambda_{1,1}^1 = 5$, $\lambda_{1,2}^0 = 10$, $\lambda_{1,2}^1 = 20$ (follower 1), $\lambda_{2,1}^0 = 3$, $\lambda_{2,1}^1 = 3$, $\lambda_{2,2}^0 = 2$, $\lambda_{2,2}^1 = 2$ (follower 2), $\lambda_{3,1}^0 = 5$, $\lambda_{3,1}^1 = 10$, $\lambda_{3,2}^0 = 6$, $\lambda_{3,2}^1 = 6$ (follower 3), $\lambda_{4,1}^0 = 10$, $\lambda_{4,1}^1 = 15$, $\lambda_{4,2}^0 = 15$, $\lambda_{4,2}^1 = 30$ (follower 4).

For distributed controller (8.32), the controller parameters are chosen the same as controller (8.11). In this case, controller (8.32) is just the reduced part by removing the disturbance compensation term $\hat{d}_i, i \in F$ from controller (8.11).

Simulation results are shown in Figs. 8.3, 8.4, 8.5 and 8.6. It can be seen from the figures that the distributed observer (8.4) and the DO (8.12) work efficiently and both of their estimation errors converge to zero quickly in a finite time. Moreover, in the presence of external disturbances, the composite distributed controller (8.11) achieves the containment control goal in a finite time, while the reduced distributed controller (8.32) fails to do this.

8.5 Conclusions

In this chapter, the finite-time containment control has been discussed for disturbed second-order multi-agent systems. To handle the external disturbances, finite-time disturbance observers have been developed for the followers. Based on the disturbance estimates and the adding a power integrator technique, composite distributed controllers have been designed. Compared with the reduced distributed controllers without the disturbance compensation terms, the proposed composite distributed controllers provide much better disturbance rejection property for the closed-loop system.

Acknowledgments This work was supported by the National Natural Science Foundation of China under Grant 61473080, the Science Foundation for Distinguished Young Scholars of Jiangsu Province under Grant BK20130018, the Priority Academic Program Development of Jiangsu Higher Education Institutions, the Fundamental Research Funds for the Central Universities, the China Postdoctoral Science Foundation under Grant 2015M570398, the Open Fund of Key Laboratory of Measurement and Control of Complex Systems of Engineering, Ministry of Education under Grant MCCSE2015B03, and the Natural Science Foundation of Jiangsu Province.

Appendix

Proof of Proposition 8.2: For brevity, denote $d_F = [d_1^T, \dots, d_n^T]^T, \hat{d}_F = [\hat{d}_1^T, \dots, \hat{d}_n^T]^T, \tilde{d}_F = [\tilde{d}_1^T, \dots, \tilde{d}_n^T]^T$. By Lemma 8.5, the DO (8.12) is finite-time convergent. So there is a constant h such that $\|\tilde{d}_i(t)\|_2 \leq h, \forall i \in F, t \in [0, +\infty)$. Define $W(x_F, v_F) = \frac{1}{2}x_F^T x_F + \frac{1}{2}v_F^T v_F$. \dot{W} along system (8.3) satisfies

$$\dot{W} = x_F^T v_F + v_F^T (u_F + d_F) \leq W + \sum_{i=1}^n \|v_i\|_2 (\|u_i^*\|_2 + h). \quad (8.34)$$

where $u_i^* = u_i + \hat{d}_i$. From (8.11), it can be obtained that

$$\|u_i^*\|_2 \leq \|\dot{v}_i^d\|_2 + k_2 \|\zeta_i^{2q-1}\|_2, \quad i \in F. \quad (8.35)$$

where $\zeta_i = (v_i - \hat{v}_i^d)^{1/q} + k_1^{1/q} \sum_{j \in F \cup L} a_{ij}(x_i - x_j)$. According to Lemma 8.2, $\forall y = [y_1, \dots, y_p]^T \in \mathbb{R}^p, a \geq 0$, the following holds

$$\|y^a\|_2 = \left[\sum_{k=1}^p (y_k^a)^2 \right]^{1/2} \leq \sum_{k=1}^p |y_k|^a \leq p \left(\sum_{k=1}^p y_k^2 \right)^{a/2} = p \|y\|_2^a. \quad (8.36)$$

Clearly, (8.36) holds by letting $y = (v_i - \hat{v}_i^d)^{1/q} + k_1^{1/q} \sum_{j \in F \cup L} a_{ij}(x_i - x_j) \in \mathbb{R}^p, a = 2q - 1$ or $y = v_i - \hat{v}_i^d \in \mathbb{R}^p, a = 1/q$. Based on the fact $0 < 2q - 1 < 1$ and Lemma 8.2, it follows that $p\|(v_i - \hat{v}_i^d)^{1/q} + k_1^{1/q} \sum_{j \in F \cup L} a_{ij}(x_i - x_j)\|_2^{2q-1} \leq p\|(v_i - \hat{v}_i^d)^{1/q}\|_2^{2q-1} + pk_1^{2-1/q} \|\sum_{j \in F \cup L} a_{ij}(x_i - x_j)\|_2^{2q-1}$. Note that $0 < 2 - 1/q < 1$. Based on (8.36) and Lemma 8.2, it can be verified that $\|(v_i - \hat{v}_i^d)^{1/q}\|_2^{2q-1} \leq p^{2q-1}(\|v_i\|_2 + \|\hat{v}_i^d\|_2)^{2-1/q} \leq p^{2q-1}(\|v_i\|_2^{2-1/q} + \|\hat{v}_i^d\|_2^{2-1/q})$. In addition, $\|\sum_{j \in F \cup L} a_{ij}(x_i - x_j)\|_2 \leq \beta \sum_{j=1}^{n+m} (\|x_i\|_2 + \|x_j\|_2) = \beta(n+m)\|x_i\|_2 + \beta \sum_{j=1}^n \|x_j\|_2 + \beta \sum_{j=n+1}^{n+m} \|x_j\|_2$, where $\beta = \max_{\forall i \in F} \left\{ \sum_{j \in F \cup L} a_{ij} \right\}$. Then it can be obtained from (8.36) that

$$\begin{aligned} p\|\zeta_i\|_2^{2q-1} &\leq p^{2q} \left(\|v_i\|_2^{2-1/q} + \|\hat{v}_i^d\|_2^{2-1/q} \right) + pk_1^{2-1/q} \beta^{2q-1} \\ &\quad \times \left[(n+m)^{2q-1} \|x_i\|_2^{2q-1} + \sum_{j=1}^n \|x_j\|_2^{2q-1} + \sum_{j=n+1}^{n+m} \|x_j\|_2^{2q-1} \right], \quad i \in F. \end{aligned} \quad (8.37)$$

Then, putting (8.35)–(8.37) together yields

$$\begin{aligned} \|u_i^*\|_2 &\leq \|\dot{v}_i^d\|_2 + k_2 p^{2q} \left(\|v_i\|_2^{2-1/q} + \|\hat{v}_i^d\|_2^{2-1/q} \right) + k_2 p k_1^{2-1/q} \beta^{2q-1} \\ &\quad \times \left[(n+m)^{2q-1} \|x_i\|_2^{2q-1} + \sum_{j=1}^n \|x_j\|_2^{2q-1} + \sum_{j=n+1}^{n+m} \|x_j\|_2^{2q-1} \right] \\ &\leq \delta_1 + k_2 p^{2q} \|v_i\|_2^{2-1/q} \\ &\quad + k_2 p k_1^{2-1/q} \beta^{2q-1} \left[(n+m)^{2q-1} \|x_i\|_2^{2q-1} + \sum_{j=1}^n \|x_j\|_2^{2q-1} \right], \quad i \in F, \end{aligned} \quad (8.38)$$

where $\delta_1 = \|\dot{v}_i^d\|_2 + k_2 p^{2q} \|\hat{v}_i^d\|_2^{2-1/q} + k_2 p k_1^{2-1/q} \beta^{2q-1} \sum_{j=n+1}^{n+m} \|x_j\|_2^{2q-1}$. Note that $\dot{v}_i^d = -\rho_1 \text{sig}^\alpha \left(\sum_{j \in F \cup L} a_{ij} (\hat{v}_i^d - \hat{v}_j^d) \right) - \rho_2 \text{sgn} \left(\sum_{j \in F \cup L} a_{ij} (\hat{v}_i^d - \hat{v}_j^d) \right)$. Due

to global convergence of observer (8.4) and Assumption 6, the existence of δ_1 is guaranteed. Then it follows from (8.34), (8.38) and Lemma 8.3 that

$$\begin{aligned} \dot{W} &\leq W + (\delta_1 + h) \sum_{i=1}^n \|v_i\|_2 + k_2 p^{2q} \sum_{i=1}^n \|v_i\|_2^{3-1/q} \\ &\quad + \frac{k_2 p k_1^{2-1/q} \beta^{2q-1}}{2q} \sum_{i=1}^n [(n+m)^{2q-1} + n] \|v_i\|_2^{2q} \\ &\quad + \frac{k_2 p k_1^{2-1/q} \beta^{2q-1} (2q-1)}{2q} \sum_{i=1}^n \left[(n+m)^{2q-1} \|x_i\|_2^{2q} + \sum_{j=1}^n \|x_j\|_2^{2q} \right] \\ &\leq W + (\delta_1 + h) \sum_{i=1}^n \|v_i\|_2 + k_2 p^{2q} \sum_{i=1}^n \|v_i\|_2^{3-1/q} \\ &\quad + \frac{k_2 p k_1^{2-1/q} \beta^{2q-1} [(n+m)^{2q-1} + n]}{2q} \sum_{i=1}^n (\|x_i\|_2^{2q} + \|v_i\|_2^{2q}). \end{aligned} \quad (8.39)$$

In addition, it holds that $\max\{\|x_i\|_2^a, \|v_i\|_2^a\} \leq (\|x_i\|_2^2 + \|v_i\|_2^2)^{a/2}$, $\forall a \geq 0$. Then based on the fact $0 < q$, $(3-1/q)/2 < 1$ and (8.39), it follows that

$$\begin{aligned} \dot{W} &\leq W + (\delta_1 + h) \sum_{i=1}^n (\|x_i\|_2^2 + \|v_i\|_2^2)^{1/2} + k_2 p^{2q} \sum_{i=1}^n (\|x_i\|_2^2 + \|v_i\|_2^2)^{(3-1/q)/2} \\ &\quad + \frac{k_2 p k_1^{2-1/q} \beta^{2q-1} [(n+m)^{2q-1} + n]}{q} \sum_{i=1}^n (\|x_i\|_2^2 + \|v_i\|_2^2)^q. \end{aligned} \quad (8.40)$$

Note that $\|x\|_p = (\sum_{l=1}^m |x_l|^p)^{1/p}$, $\forall x = [x_1, \dots, x_m]^T$ with $p \geq 1, m \in N^+$ denotes p -norm in \mathbb{R}^m . Based on the equivalence between any two different norms in \mathbb{R}^p and Lemma 8.2, there is $\delta_2 > 0$ such that $\sum_{i=1}^n (\|x_i\|_2^2 + \|v_i\|_2^2)^{1/2} \leq \sum_{i=1}^n (\|x_i\|_2 + \|v_i\|_2) \leq \delta_2 W^{1/2}$. Similarly, both $\sum_{i=1}^n (\|x_i\|_2^2 + \|v_i\|_2^2)^{(3-1/q)/2} \leq \delta_3 W^{(3-1/q)/2}$ and $\sum_{i=1}^n (\|x_i\|_2^2 + \|v_i\|_2^2)^q \leq \delta_4 W^q$ hold with appropriate $\delta_3 > 0, \delta_4 > 0$. Then it follows from (8.40) that

$$\begin{aligned} \dot{W} &\leq W + (\delta_1 + h) \delta_2 W^{1/2} + k_2 p^{2p} \delta_3 W^{(3-1/q)/2} \\ &\quad + \frac{k_2 p k_1^{2-1/q} \beta^{2q-1} [(n+m)^{2q-1} + n] \delta_4}{q} W^q. \end{aligned} \quad (8.41)$$

By Lemma 8.3, it holds that $W^b = W^b \cdot 1^{1-b} \leq bW + 1 - b$, $\forall 0 < b \leq 1$. Then it can be obtained from (8.41) that

$$\dot{W} \leq \delta_5 W + \delta_6, \quad (8.42)$$

where $\delta_5 = 1 + \frac{(\delta_1+h)\delta_2}{2} + \frac{3q-1}{2q}k_2p^{2q}\delta_3 + k_2pk_1^{2-1/q}\beta^{2q-1}[(n+m)^{2q-1}+n]\delta_4$ and $\delta_6 = \frac{(\delta_1+h)\delta_2}{2} + \frac{1-q}{2q}k_2p^{2q}\delta_3 + \frac{k_2pk_1^{2-1/q}\beta^{2q-1}[(n+m)^{2q-1}+n]\delta_4(1-q)}{q}$. By noting that $\delta_5, \delta_6 \in (0, +\infty)$, it follows from (8.42) that W is bounded, which implies that $x_i(t), v_i(t), i \in F$ are bounded $\forall t \in [0, +\infty)$. This completes the proof. \square

References

1. Ou, M., Du, H., Li, S.: Finite-time formation control of multiple nonholonomic mobile robots. *Int. J. Robust Nonlinear Control* **24**(1), 140–165 (2014)
2. Dong, X., Yu, B., Shi, Z., Zhong, Y.: Time-varying formation control for unmanned aerial vehicles: theories and applications. *IEEE Trans. Control Syst. Technol.* **23**(1), 340–348 (2015)
3. Li, Z., Li, J., Kang, Y.: Adaptive robust coordinated control of multiple mobile manipulators interacting with rigid environments. *Automatica* **46**(12), 2028–2034 (2010)
4. Zhang, W., Feng, G., Yu, L.: Multi-rate distributed fusion estimation for sensor networks with packet losses. *Automatica* **48**(9), 2016–2028 (2012)
5. Pettersen, K.Y., Gravdahl, J.T., Nijmeijer, H.: *Group Coordination and Cooperative Control*. Springer, Berlin (2006)
6. Lü, J., Chen, G.: A time-varying complex dynamical network model and its controlled synchronization criteria. *IEEE Trans. Autom. Control* **50**(6), 841–846 (2005)
7. Zhou, J., Lu, J., Lü, J.: Adaptive synchronization of an uncertain complex dynamical network. *IEEE Trans. Autom. Control* **51**(4), 652–656 (2006)
8. Olfati-Saber, R., Murray, R.M.: Consensus problems in networks of agents with switching topology and time-delays. *IEEE Trans. Autom. Control* **49**(9), 1520–1533 (2004)
9. Hong, Y., Hu, J., Gao, L.: Tracking control for multi-agent consensus with an active leader and variable topology. *Automatica* **42**(7), 1177–1182 (2006)
10. Chen, Y., Lü, J., Han, F., Yu, X.: On the cluster consensus of discrete-time multi-agent systems. *Syst. Control Lett.* **60**(7), 517–523 (2011)
11. Ren, W.: On consensus algorithms for double-integrator dynamics. *IEEE Trans. Autom. Control* **53**(6), 1503–1509 (2008)
12. Yu, W., Ren, W., Zheng, W.X., Chen, G., Lü, J.: Distributed control gains design for consensus in multi-agent systems with second-order nonlinear dynamics. *Automatica* **49**(7), 2107–2115 (2013)
13. Li, S., Du, H., Lin, X.: Finite-time consensus algorithm for multi-agent systems with double-integrator dynamics. *Automatica* **47**(8), 1706–1712 (2011)
14. Li, S., Wang, X.: Finite-time consensus and collision avoidance control algorithms for multiple AUVs. *Automatica* **49**(11), 3359–3367 (2013)
15. Tian, Y., Zhang, Y.: High-order consensus of heterogeneous multi-agent systems with unknown communication delays. *Automatica* **48**(6), 1205–1212 (2012)
16. Khoo, S., Xie, L., Zhao, S., Man, Z.: Multi-surface sliding control for fast finite-time leader-follower consensus with high order SISO uncertain nonlinear agents. *Int. J. Robust Nonlinear Control* **24**(16), 2388–2404 (2014)
17. Zhang, X., Liu, L., Feng, G.: Leader-follower consensus of time-varying nonlinear multi-agent systems. *Automatica* **52**, 8–14 (2015)
18. Couzin, I.D., Krause, J., Franks, N.R., Levin, S.A.: Effective leadership and decision-making in animal groups on the move. *Nature* **433**(3), 513–516 (2005)
19. Notarstefano, G., Egerstedt, M., Haque, M.: Containment in leader-follower networks with switching communication topologies. *Automatica* **47**(5), 1035–1040 (2011)
20. Liu, H., Xie, G., Wang, L.: Necessary and sufficient conditions for containment control of networked multi-agent systems. *Automatica* **48**(7), 1415–1422 (2012)

21. Lou, Y., Hong, Y.: Target containment control of multi-agent systems with random switching interconnection topologies. *Automatica* **48**(5), 879–885 (2012)
22. Li, Z., Ren, W., Liu, X., Fu, M.: Distributed containment control of multi-agent systems with general linear dynamics in the presence of multiple leaders. *Int. J. Robust Nonlinear Control* **23**(5), 534–547 (2013)
23. Wen, G., Duan, Z., Zhao, Y., Yu, W., Cao, J.: Robust containment tracking of uncertain linear multi-agent systems: a non-smooth control approach. *Int. J. Control* **87**(12), 2522–2534 (2014)
24. Bhat, S.P., Bernstein, D.S.: Finite-time stability of continuous autonomous systems. *SIAM J. Control Optim.* **38**(3), 751–766 (2000)
25. Ding, S., Li, S.: Stabilization of the attitude of a rigid spacecraft with external disturbances using finite-time control techniques. *Aerospace Sci. Technol.* **13**(4), 256–265 (2009)
26. Meng, Z., Ren, W., You, Z.: Distributed finite-time attitude containment control for multiple rigid bodies. *Automatica* **46**(12), 2092–2099 (2010)
27. Cao, Y., Stuart, D., Ren, W., Meng, Z.: Distributed containment control for multiple autonomous vehicles with double-integrator dynamics: algorithms and experiments. *IEEE Trans. Control Syst. Technol.* **19**(4), 929–938 (2011)
28. He, X., Wang, Q., Yu, W.: Finite-time containment control for second-order multiagent systems under directed topology. *IEEE Trans. Circuits Syst. II: Express Br.* **61**(8), 619–623 (2014)
29. Wang, X., Li, S., Shi, P.: Distributed finite-time containment control for double-integrator multi-agent systems. *IEEE Trans. Cybern.* **44**(9), 1518–1528 (2014)
30. Li, S., Yang, J., Chen, W.-H., Chen, X.: *Disturbance Observer-Based Control: Methods and Applications*. CRC Press, Boca Raton (2014)
31. Chen, W.-H.: Disturbance observer based control for nonlinear system. *IEEE/ASME Trans. Mechatron.* **9**(4), 706–710 (2004)
32. Guo, L., Chen, W.-H.: Disturbance attenuation and rejection for systems with nonlinearity via DOBC approach. *Int. J. Robust Nonlinear Control* **15**(3), 109–125 (2005)
33. Kim, W., Shin, D., Choo, C.C.: Microstepping using a disturbance observer and a variable structure controller for permanent-magnet stepper motors. *IEEE Trans. Ind. Electron.* **60**(7), 2689–2699 (2013)
34. Yang, J., Li, S., Yu, X.: Sliding-mode control for systems with mismatched uncertainties via a disturbance observer. *IEEE Trans. Ind. Electron.* **60**(1), 160–169 (2013)
35. Ginoya, D., Shendge, P.D., Phadke, S.B.: Sliding mode control for mismatched uncertain systems using an extended disturbance observer. *IEEE Trans. Ind. Electron.* **61**(4), 1983–1992 (2014)
36. Li, S., Sun, H., Yang, J., Yu, X.: Continuous finite-time output regulation for disturbed systems under mismatching condition. *IEEE Trans. Autom. Control* **60**(1), 277–282 (2015)
37. Yang, H., Zhang, Z., Zhang, S.: Consensus of second-order multi-agent systems with exogenous disturbances. *Int. J. Robust Nonlinear Control* **21**(9), 945–956 (2011)
38. Zhang, X., Liu, X.: Further results on consensus of second-order multi-agent systems with exogenous disturbance. *IEEE Trans. Circuits Syst. I: Regul. Papers* **60**(12), 3215–3226 (2013)
39. Huang, X., Lin, W., Yang, B.: Global finite-time stabilization of a class of uncertain nonlinear systems. *Automatica* **41**(5), 881–888 (2005)
40. Graham, A.: Kronecker products and matrix calculus with applications. Halsted Press, New York (1981)
41. Shtessel, Y.B., Shkolnikov, I.A., Levant, A.: Smooth second-order sliding modes: Missile guidance application. *Automatica* **43**(8), 1470–1476 (2007)
42. Qian, C., Lin, W.: A continuous feedback approach to global strong stabilization of nonlinear systems. *IEEE Trans. Autom. Control* **46**(7), 1061–1079 (2001)

Chapter 9

Application of Fractional-Order Calculus in a Class of Multi-agent Systems

Wenwu Yu, Guanghui Wen and Yang Li

Abstract This chapter is concerned with fractional-order consensus problem in multi-agent systems. A brief introduction of fractional-order calculus is given in Sect. 9.1. The design of observer for consensus of a linear fractional-order multi-agent system is discussed in Sect. 9.2. Section 9.3 considers a multi-agent system consisting of second-order leader and fractional-order followers where a necessary and sufficient condition of tracking consensus is derived by using only the relative local position information of neighboring agents. The stabilization consensus problem of uncertain fractional-order multi-agent system is investigated in Sect. 9.4.

Coordination of multi-agent systems has numerous applications. Examples include flocking, swarming, formation control, and sensor networks [1–4]. The study within this field focuses on analyzing how globally coordinative group behavior emerges as a result of local interactions among the individuals. In many cooperative multi-agent systems, a group of agents only share information with their neighbors locally and simultaneously try to agree on certain global criteria of common interest.

Recently, consensus, one of the most typical collective behaviors, plays an important role in the distributed coordination which usually refers to the problem of how to reach an agreement, such as the positions, velocities, and attitudes, among a group of autonomous mobile agents in a dynamical system [5–9]. Recently, second-order consensus problem has come to be an important topic [10–15], where each agent is governed by second-order dynamics. In general, the second-order consensus problem refers to the problem of reaching an agreement among a group of autonomous agents governed by second-order dynamics. Some sufficient conditions were derived for reaching second-order consensus in linear models [12, 14]. In [14], some necessary and sufficient conditions were obtained for second-order consensus in a linear multi-agent system containing a directed spanning tree with or without delay. It was found that both the real and imaginary parts of the eigenvalues of Laplacian matrix of the network play key roles in reaching second-order consensus in general. In

W. Yu (✉) · G. Wen · Y. Li

Department of Mathematics, Southeast University, Nanjing 210096, Jiangsu Province,
People's Republic of China
e-mail: wwwu@seu.edu.cn

G. Wen
e-mail: wenguanghui@gmail.com

the leader-following case, each agent should exchange both position and velocity information with its neighbors in order to reach second-order tracking consensus.

Fractional calculus can be dated back to the seventeenth century. Different from the integer orders of derivatives and integrals in the conventional calculus, the orders of derivatives and integrals in fractional calculus can be any given positive real numbers. It should be noted that, to date, most papers have studied integer-order multi-agent systems. However, it has been pointed out by many researchers that many physical systems are more suitable to be described by fractional-order dynamic equations rather than by the classic integer-order ones, such as, vehicles moving on top of viscoelastic materials (e.g., sand, muddy road) or high-speed aircrafts traveling in an environment with the influence of particles (e.g., rain, snow) [16]. Moreover, many phenomena can be explained naturally by the collective group behavior of agents with fractional-order dynamics. For example, the synchronized motion of agents in fractional circumstances, such as macromolecule fluids and porous media. Fractional-order derivatives provide an excellent instrument for the description of memories which are neglected in the classical integer-order models. In addition, fractional-order systems include traditional integer-order systems as special cases.

Most of the real systems can be described by fractional-order systems essentially [17–21]. In the past, the integer-order systems were used to describe the nature. But in recent years, it was found that the traditional integer-order differential equation can not be used to accurately describe many phenomena in nature, while the fractional-order system is able to accumulate a certain range of all the information with good memories and thus has wider applications. This provides an excellent tool for the description of memories which are often neglected in the classical integer-order models. Thus it is more reasonable to model the practical systems by fractional-order systems. The calculation of fractional order in many areas of science and engineering plays an important role, which has also attracted increasing attention of many researchers from different fields.

Sometimes, the state information of agent cannot be measured directly in multi-agent systems, so the observer-based control laws depending on the output measurements of the agents are needed [22–24]. The investigation on consensus of fractional-order systems meets certain difficulties for lacking the similar stability theory of integer-order ones. The typically used methods are Laplace transforms and numerical simulations, which all have some limitations in dealing with the large-scale systems. In addition, the system's evolution will be inevitably affected by external disturbance and uncertain parameters in most practical systems. Thus, it is important to include the uncertainties or disturbances into the model of considered system. Based on the above discussions, this chapter mainly investigates the applications of fractional-order calculus in multi-agent cooperative control problem by using the properties of fractional-order calculus. The main contribution is that some new control protocols are derived to reach consensus in a nominal multi-agent system or stabilization consensus in a class of uncertain fractional-order multi-agent systems.

9.1 Preliminaries

The definition of fractional calculus is an expansion of fractional calculus, and fractional differential equation theory is actually an expansion of integer order differential equation theory. Meanwhile, the numerical method for solving the fractional order differential equation is still an important research topic in the modern mathematics. All the related theories provide a solid theoretical basis of fractional-order system theory. Some preliminaries on fractional calculus are first presented in this section. In addition, algebraic graph theory which is used in analyzing the cooperative behavior of multi-agent systems will also be introduced in this section.

9.1.1 Introduction of Some Basical Functions

Gamma function, Bata function and Mittag-Leffler function are basic functions which are widely used in the definition of fractional calculus. However, there is no unified definition about fractional order calculus until now which is the main reason why the fractional order calculus has not been widely applied in practice. Before giving the definition of fractional calculus, here we first introduce three classical functions: Gamma function, Bata function and Mittag-Leffler function.

1. Gamma Function

The definition $\Gamma(z)$ is defined as follows:

$$\Gamma(z) = \int_0^\infty e^{-t} t^{z-1} dt,$$

where z is a complex number located at the right half plane of complex plane, that is $Re(z) > 0$.

One property of Gamma function can be verified by:

$$\Gamma(z) = \lim_{n \rightarrow \infty} \frac{n! n^z}{z(z+1)\cdots(z+n)},$$

where n is a complex number. By the above definition, the following property of Gamma function holds

$$\Gamma(z+1) = z\Gamma(z).$$

2. Bata Function

Bata function can be expressed as

$$B(z, \omega) = \int_0^1 \tau^{z-1} (1-\tau)^{\omega-1} d\tau, \quad Re(z) > 0, Re(\omega) > 0.$$

We can use the Laplace transform to build the relationship between Bata function and Gamma function. It is convenient in some cases to represent the Gamma function by using the Bata function. By Laplace transform, we have

$$B(z, \omega) = \frac{\Gamma(z)\Gamma(\omega)}{\Gamma(z + \omega)}.$$

3. Mittag-Leffler Function

Mittag-Leffler function with one parameter is given by

$$E_\alpha(z) = \sum_{k=0}^{\infty} \frac{z^k}{\Gamma(\alpha k + 1)}, \quad \alpha > 0.$$

Mittag-Leffler function with two parameters is defined as

$$E_{\alpha, \beta}(z) = \sum_{k=0}^{\infty} \frac{z^k}{\Gamma(\alpha k + \beta)}, \quad \alpha > 0, \quad \beta > 0.$$

The Mittag-Leffler function with one parameter can be seen as a special case of the one with two parameters. The Laplace transform of Mittag-Leffler function has simple closed form, i.e.,

$$\int_0^\infty e^{-pt} t^{\alpha k + \beta - 1} E_{\alpha, \beta}^{(k)}(\pm at^\alpha) dt = \frac{k! p^{\alpha - \beta}}{(p^\alpha \mp a)^{k+1}}, \quad Re(p) > |a|^{1/\alpha}.$$

9.1.2 Definition of Fractional Calculus

Symbol ${}_a D_t^\alpha$ is the basic operator of fractional order calculus, where a and t are the upper and lower limits of operator respectively, and α is the fractional order. Differential and integral operator can be unified together through the basic operation of fractional calculus. Currently, three kinds of definitions are commonly used: Grunwald-Letnikov fractional calculus, Riemann-Liouville fractional calculus and Caputo fractional calculus. These three definitions are provided from different aspects but they do have certain relationship between each other. Here is a brief introduction of the three different definitions of fractional order calculus. Firstly, the basic definition of fractional calculus is given:

Definition 9.1 [20, 21] The α -order integration of function $f(t) : D_{t_0, t}^\alpha f(t)$ is

$$D_{t_0, t}^\alpha f(t) = \frac{1}{\Gamma(\alpha)} \int_{t_0}^t (t - \tau)^{\alpha-1} f(\tau) d\tau,$$

where $f(t)$ is the function of time, α is the fractional order, t_0 stands for the initial time, and $\Gamma(\cdot)$ represents the Gamma function.

Based on the above definition, the definitions of Grunwald-Letnikov fractional calculus, Riemann-Liouville fractional calculus and Caputo fractional calculus are introduced as follows.

1. Grunwald-Letnikov fractional calculus

For any real number m , let $[m]$ be the integer part of m . Then, the α -order fractional calculus of $f(t)$ is defined as follows.

Definition 9.2 [20, 21]

$$\begin{aligned} {}_aD_t^\alpha f(t) &= \lim_{h \rightarrow 0} h^{-\alpha} \sum_{j=0}^{[(t-\alpha)/h]} (-1)^j \frac{\alpha!}{j!(\alpha-j)!} f(t-jh) \\ &= \lim_{h \rightarrow 0} \frac{1}{\Gamma(\alpha)h^\alpha} \sum_{j=0}^{[(t-\alpha)/h]} \frac{\Gamma(\alpha+j)}{\Gamma(j+1)} f(t-jh). \end{aligned}$$

2. Riemann-Liouville fractional calculus

Definition 9.3 [20, 21] Riemann-Liouville differential of function $f(t) : {}_{RL}D_{t_0,t}^\alpha f(t)$ is

$$\begin{aligned} {}_{RL}D_{t_0,t}^\alpha f(t) &= \frac{1}{\Gamma(n-\alpha)} \frac{d^n}{dt^n} \int_{t_0}^t (t-\tau)^{n-\alpha-1} f(\tau) d\tau \\ &= \frac{d^n}{dt^n} D_{t_0,t}^{-(n-\alpha)} f(t) \end{aligned}$$

where $n-1 < \alpha \leq n$.

3. Caputo fractional calculus

Definition 9.4 [20, 21] Caputo differential of function $f(t) : {}_C D_{t_0,t}^\alpha f(t)$ is

$$\begin{aligned} {}_C D_{t_0,t}^\alpha f(t) &= \frac{1}{\Gamma(n-\alpha)} \int_{t_0}^t (t-\tau)^{n-\alpha-1} f^{(n)}(\tau) d\tau \\ &= D_{t_0,t}^{-(n-\alpha)} \frac{d^n}{dt^n} f(t) \end{aligned}$$

where $n-1 < \alpha \leq n$.

Riemann-Liouville fractional calculus and Caputo fractional calculus are both improvements of Grunwald-Letnikov fractional calculus. Riemann-Liouville fractional order calculus can be simplified to the calculation of fractional-order derivative. The introduction of Caputo fractional calculus simplifies the Laplace transform which makes the discussion about fractional-order differential equations become

much easy. In addition, Riemann-Liouville fractional-order differential operation involves the issue of the initial value while Caputo fractional-order differential operation has the same form of disposal conditions with integral-order differential operation which makes the Caputo fractional-order calculus have more wide applications than Riemann-Liouville-type calculus in practical systems. Besides, the Laplace transform of Caputo fractional calculus is simpler than Riemann-Liouville-type, so the Caputo fractional calculus will be used throughout the whole chapter. For convenience, we take $t_0 = 0$ and use $f^{(\alpha)}(t)$ to represent ${}_C D_{t_0, t}^\alpha f(t)$.

Then, a brief introduction about the Laplace transform of Caputo fractional order differential is given as follows. Let $\mathcal{L}(\cdot)$ be the Laplace transform. Based on the definition of Laplace transform, we have:

$$\mathcal{L}(f^\alpha(t)) = \begin{cases} s^\alpha F(s) - s^{\alpha-1} f(0), & \alpha \in (0, 1], \\ s^\alpha F(s) - s^{\alpha-1} f(0) - s^{\alpha-2} \dot{f}(0), & \alpha \in (1, 2]. \end{cases}$$

Before moving on, the following lemmas are presented.

Lemma 9.1 [20, 21] *Fractional-order system*

$$D^\alpha x(t) = Ax(t), \quad 0 < \alpha < 1,$$

is asymptotically stable if there exists a matrix $P > 0$ with an appropriate dimension, such that

$$(A^{\frac{1}{\alpha}})^T P + P(A^{\frac{1}{\alpha}}) < 0.$$

Lemma 9.2 [20, 21] *Autonomous system*

$$D^\alpha x(t) = Ax(t), \quad x(t_0) = x_0, \quad 0 < \alpha < 1,$$

is asymptotically stable if and only if

$$|\arg(\text{spec}(A))| > \alpha \frac{\pi}{2},$$

where $\text{spec}(A)$ is the spectrum of A .

9.1.3 Algebraic Graph Theory

Let $R^{n \times n}$ and $C^{n \times n}$ be the sets of $n \times n$ real and complex matrices, respectively. I denotes the identity matrix with an appropriate dimension. Denote $\mathbf{1}$ by the column vector with all entries being equal to one. The superscript T means the transpose for real matrices and $*$ means the conjugate transpose for complex matrices. α denotes the fractional order. In this section, it is assumed that $0 < \alpha < 1$.

Notation $\text{diag}(A_1, A_2, \dots, A_n)$ represents a block-diagonal matrix with matrices $A_i, i = 1, 2, \dots, n$, on its diagonal part. The matrix inequality $A < 0$ means A is a negative definite matrix. $\bar{\sigma}(A)$ denotes the maximal singular value of matrix A , and $A \otimes B$ denotes the Kronecker product of matrices A and B . Furthermore, the Kronecker product has the following properties:

Lemma 9.3 [25] For matrix A, B, C and D with appropriate dimensions,

- (1) $(\gamma A) \otimes B = A \otimes (\gamma B)$,
- (2) $(A + B) \otimes C = A \otimes C + B \otimes C$,
- (3) $(A \otimes B)(C \otimes D) = (AC) \otimes (BD)$,
- (4) $(A \otimes B)^T = A^T \otimes B^T$,

where γ is a constant.

For a system consisting of N agents, the interaction graph for all agents can be modeled by a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, 2, \dots, N\}$ and $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ represent the node set and the edge set of the graph, respectively [26]. An edge is denoted by a pair of distinct nodes of $\mathcal{G} : (i, j) \in \mathcal{E}$, which means node j can access to the state information of node i . Accordingly, node i is a neighbor of node j . All neighbors of node i are denoted by \mathcal{N}_i . A path on \mathcal{G} from nodes i_1 to i_k is a sequence of ordered edges in the form of $(i_j, i_{j+1}) \in \mathcal{E}, j = 1, 2, \dots, k - 1$. A directed graph has or contains a directed spanning tree if there exists a node called root such that there exists a directed path from this node to every other nodes. $\arg(\cdot)$ denotes the argument of a complex number and takes value in $[-\pi, \pi]$, $i = 1, 2, \dots, N$.

Suppose that there are N nodes in graph \mathcal{G} . The adjacency matrix of graph \mathcal{G} is denoted by $A = [a_{ij}] \in R^{N \times N}$, in which $a_{ij} = 1$ if $(j, i) \in \mathcal{E}$ and $a_{ij} = 0$ otherwise. Assume that there are no self-loops, i.e. $a_{ii} = 0$ for $i \in \mathcal{V}$. The Laplacian matrix $L = [L_{ij}] \in R^{N \times N}$ of graph \mathcal{G} is defined by $L_{ii} = \sum_{j \neq i} a_{ij}$, $L_{ij} = -a_{ij}$ for $i \neq j$, which ensures the diffusion property $\sum_{j=1}^N L_{ij} = 0$.

Lemma 9.4 [15] Suppose that graph \mathcal{G} has a directed spanning tree. Then, L has a simple zero eigenvalue and all the other eigenvalues have positive real parts.

9.2 Observer Design for Consensus of General Fractional-Order Multi-agent Systems

Since under many circumstances, some states of agents can not be directly measured [22–24]. This motivates the studies of the current section. A fractional-order observer-type consensus protocol for a general multi-agent system is proposed in this section. Combined with LMI approach and matrix decomposition method, consensus problem of the original system can be transformed into the stability problem of a set of matrices which reduces the dimension of original system and computational

complexity compared with directly analyzing the system by using general Nyquist stability criterion.

9.2.1 Main Results

Consider a group of N identical agents with general linear fractional-order dynamics. The dynamics of the i th agent are described by

$$x_i^{(\alpha)} = Ax_i + Bu_i, \quad y_i = Cx_i, \quad (9.1)$$

where $x_i = [x_{i1}, x_{i2}, \dots, x_{in}]^T \in R^n$ is the state of the i th agent, $u_i \in R^p$ is the control input, and $y_i \in R^q$ is the measured output.

It is assumed that the communication topology among the agents is described by a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with the set of agents $\mathcal{V} = \{1, 2, \dots, N\}$ and the set of edges $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$. An edge $(i, j) \in \mathcal{E}$ means agent j can obtain information from agent i , but not vice versa.

The measurements of other agents with respect to agent i can be synthesized into a single signal as follows:

$$\xi_i = c \sum_{j=1}^N a_{ij} (y_j - y_i), \quad (9.2)$$

where $c > 0$ is the coupling strength, $a_{ii} = 0$, $a_{ij} = 1(i \neq j)$ if $(j, i) \in \mathcal{E}$ and $a_{ij} = 0$, otherwise. A fractional-order observer-type consensus protocol is proposed in the following way:

$$\begin{aligned} v_i^{(\alpha)} &= (A + BK)v_i + F \left(c \sum_{j=1}^N a_{ij}C(v_j - v_i) - \xi_i \right), \\ u_i &= Kv_i, \end{aligned} \quad (9.3)$$

where $v_i \in R^n$ is the protocol state, $i = 1, 2, \dots, N$, and $F \in R^{q \times n}$ and $K \in R^{p \times n}$ are the feedback gain matrices to be determined. The term $\sum_{j=1}^N a_{ij}C(v_i - v_j)$ in (9.3) represents the information exchanges between agent i and its neighbors.

Under (9.3), system (9.1) can be rewritten as

$$\dot{\xi}_i^{(\alpha)} = \mathcal{A}\xi_i + c \sum_{j=1}^N L_{ij}\mathcal{H}\xi_j, \quad i = 1, 2, \dots, N, \quad (9.4)$$

where

$$\xi_i = \begin{pmatrix} x_i \\ v_i \end{pmatrix}, \mathcal{A} = \begin{pmatrix} A & BK \\ 0 & A + BK \end{pmatrix}, \mathcal{H} = \begin{pmatrix} 0 & 0 \\ -FC & FC \end{pmatrix},$$

$L \in R^{N \times N}$ is the Laplacian matrix of \mathcal{G} , $R^{N \times N}$ represents the set of $N \times N$ real matrices and \mathcal{H} represents the inner linking matrix.

Protocol (9.3) solves the consensus problem if the states of system (9.4) satisfy

$$\lim_{t \rightarrow \infty} \| \xi_i(t) - \xi_j(t) \| = 0, \quad \forall i, j = 1, 2, \dots, N. \quad (9.5)$$

If the states of system (9.4) satisfy (9.5), then

$$\begin{aligned} \lim_{t \rightarrow \infty} \| x_i(t) - x_j(t) \| &= 0, \\ \lim_{t \rightarrow \infty} \| v_i(t) - v_j(t) \| &= 0, \quad \forall i, j = 1, 2, \dots, N. \end{aligned}$$

It is easy to see that system (9.4) can achieve consensus if both the agent states and protocol states reach common values respectively.

Let $r^T = [r_1, r_2, \dots, r_N] \in R^{1 \times N}$ be the left eigenvector of L associated with zero eigenvalue which satisfies $r^T \mathbf{1} = 1$. Consider the error states as follows:

$$\delta(t) = \xi(t) - ((\mathbf{1} r^T) \otimes I_{2n}) \xi(t), \quad (9.6)$$

where $\xi = [\xi_1^T, \xi_2^T, \dots, \xi_N^T]^T$ and $\delta \in R^{2Nn \times 2Nn}$. It is easy to see that δ satisfies $(r^T \otimes I_{2n}) \delta = 0$. Here, δ is referred as the disagreement vector and evolves in the following disagreement dynamics:

$$\delta^{(\alpha)} = (I_N \otimes \mathcal{A} + cL \otimes \mathcal{H}) \delta. \quad (9.7)$$

Lemma 9.5 *Consensus problem of network (9.4) is equivalent to the asymptotic stability problem of the zero equilibrium point of disagreement dynamics (9.7).*

Proof The error states (9.6) can be rewritten as

$$\delta = (\Gamma \otimes I_{xn}) \xi, \quad (9.8)$$

where

$$\Gamma = I_N - \mathbf{1} r^T = \begin{pmatrix} 1 - r_1 & -r_2 & \cdots & -r_N \\ -r_1 & 1 - r_2 & \cdots & -r_N \\ \vdots & \vdots & \ddots & \vdots \\ -r_1 & -r_2 & \cdots & 1 - r_N \end{pmatrix}.$$

By the definition of r , Γ has a simple eigenvalue zero with $\mathbf{1}$ as the corresponding right eigenvector and another eigenvalue one with multiplicity $N - 1$. It is easy to see from

(9.8) through a simple computation that $\delta = 0$ if and only if $\xi_1 = \xi_2 = \dots = \xi_N$. Thus, the consensus problem of network (9.4) is solved if and only if $\delta(t) \rightarrow 0$ as $t \rightarrow \infty$.

Remark 9.1 The proposed consensus protocol (9.3) is partly motivated by [5] which is designed for integer-order multi-agent systems. However, the system in this section is fractional-order, where the analysis is more difficult than the integer-order protocol in [5]. It could be noted that under protocol (9.3), system (9.1) can be rewritten in a compact form which can be solved by using LMI approach.

In the following, we will derive a sufficient condition to ensure consensus under protocol (9.3) by using LMI approach where α satisfies $0 < \alpha < 1$.

Theorem 9.1 *For a directed network of agents with communication topology \mathcal{G} that has a directed spanning tree, protocol (9.3) with $0 < \alpha < 1$ solves the consensus problem of multi-agent system (9.1) if there exist matrices $P > 0$, $P \in R^{2Nn \times 2Nn}$, such that*

$$(\tilde{L}_i^{\frac{1}{\alpha}})^{TP} + P(\tilde{L}_i^{\frac{1}{\alpha}}) < 0, \quad i = 2, \dots, N, \quad (9.9)$$

where $\tilde{L}_i = \mathcal{A} + c\lambda_i \mathcal{H}$, in which λ_i , $i = 2, \dots, N$ are nonzero eigenvalues of the Laplacian matrix L .

Proof From Lemma 9.2, the disagreement dynamics (9.7) can achieve asymptotic stability if

$$(\mathcal{E}^{\frac{1}{\alpha}})^{TP} + P(\mathcal{E}^{\frac{1}{\alpha}}) < 0, \quad (9.10)$$

where $\mathcal{E} = I_N \otimes \mathcal{A} + cL \otimes \mathcal{H}$.

Let $U \in R^{N \times (N-1)}$, $V \in R^{(N-1) \times N}$, and $T \in C^{N \times N}$ satisfying

$$T = (\mathbf{1} \ U), \quad T^{-1} = \begin{pmatrix} r^T \\ V \end{pmatrix}, \quad \text{and} \quad T^{-1}LT = J = \begin{pmatrix} 0 & 0 \\ 0 & \Lambda \end{pmatrix}, \quad (9.11)$$

where Λ is a upper triangular matrix with diagonal entries being the nonzero eigenvalues of L . From the state transformation $\varepsilon = (T^{-1} \otimes I_{2n})\delta$ and $\varepsilon = [\varepsilon_1^T, \varepsilon_2^T, \dots, \varepsilon_N^T]^T$, dynamics (9.7) can be rewritten in terms of ε as follows:

$$\varepsilon^{(\alpha)} = (I_N \otimes \mathcal{A} + cJ \otimes \mathcal{H})\varepsilon. \quad (9.12)$$

It is easy to see from (9.6) that

$$\varepsilon_1 = (r^T \otimes I_{2n})\delta \equiv 0. \quad (9.13)$$

Since the elements of the state matrix of (9.12) are either block diagonal or block upper triangular, ε_i , asymptotically converges to zero if and only if $N - 1$ subsystems along the diagonal, i.e.,

$$\varepsilon_i^{(\alpha)} = (\mathcal{A} + c\lambda_i \mathcal{H})\varepsilon_i, \quad i = 2, \dots, N, \quad (9.14)$$

are asymptotically stable. Thus (9.10) can be rewritten as

$$(\tilde{L}_i^{\frac{1}{\alpha}})^T P + P(\tilde{L}_i^{\frac{1}{\alpha}}) < 0, \quad i = 2, \dots, N, \quad (9.15)$$

where $\tilde{L}_i = \mathcal{A} + c\lambda_i \mathcal{H}$, in which $\lambda_i, i = 2, \dots, N$ are nonzero eigenvalues of the Laplacian matrix L .

By using the similar method, a necessary and sufficient condition to ensure consensus of system (9.1) under protocol (9.3) with $0 < \alpha < 1$ is given as follows.

Theorem 9.2 *For a directed network of agents with communication topology \mathcal{G} containing a directed spanning tree, protocol (9.3) solves the consensus problem of multi-agent system (9.1) if and only if*

$$\min\{|\arg(\text{spec}(A + c\lambda_i FC))|, |\arg(\text{spec}(A + BK))|\} > \alpha \frac{\pi}{2}, \quad (9.16)$$

where $\lambda_i, i = 2, \dots, N$ are the nonzero eigenvalues of the Laplacian matrix L of \mathcal{G} .

Proof From Lemma 9.2, it is easy to know that the disagreement dynamics (9.7) can achieve asymptotical stability if and only if

$$|\arg(\text{spec}(\mathcal{E}))| > \alpha \frac{\pi}{2}, \quad (9.17)$$

where the definition of \mathcal{E} is same as (9.10).

It has been proved in Theorem 9.1 that after introducing the transformation, the asymptotic stability of (9.7) is equivalent to the asymptotic stability of (9.14). Since

$$\mathcal{A} + c\lambda_i \mathcal{H} = \begin{pmatrix} A & BK \\ -c\lambda_i FC & A + BK + c\lambda_i FC \end{pmatrix}, \quad (9.18)$$

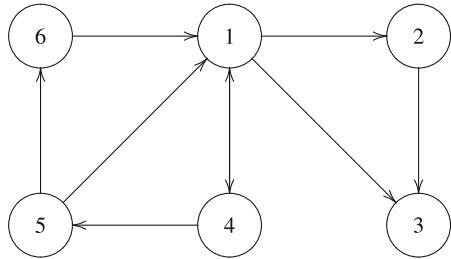
and there exists a nonsingular matrix $Q = \begin{pmatrix} I & I \\ 0 & I \end{pmatrix}$ such that

$$Q^{-1}(\mathcal{A} + c\lambda_i \mathcal{H})Q = \begin{pmatrix} A + c\lambda_i FC & 0 \\ -c\lambda_i FC & A + BK \end{pmatrix}, \quad (9.19)$$

where $I \in R^{n \times n}$ is an identical matrix. It can be seen that matrices $\mathcal{A} + c\lambda_i \mathcal{H}$ are similar to

$$\begin{pmatrix} A + c\lambda_i FC & 0 \\ -c\lambda_i FC & A + BK \end{pmatrix}, \quad i = 2, \dots, N,$$

Fig. 9.1 The communication topology



where their eigenvalues of the matrix are determined by those of their diagonal element matrices. Thus, (9.16) can be rewritten as

$$\min\{|\arg(\text{spec}(A + c\lambda_i FC))|, |\arg(\text{spec}(A + BK))|\} > \alpha \frac{\pi}{2}. \quad (9.20)$$

Remark 9.2 It can be seen from Theorem 9.2 that under the fractional-order observer-type consensus protocol, consensus of a large scale system (9.1) can be converted to the stability of a set of matrices which have the same dimension as a single agent. This method reduces the computational complexity greatly which is also the key important step of this theorem.

9.2.2 Numerical Simulation

In this subsection, an example is given to verify the effectiveness of the theoretical results.

Assume that the communication topology is shown in Fig. 9.1, so the corresponding Laplacian matrix is

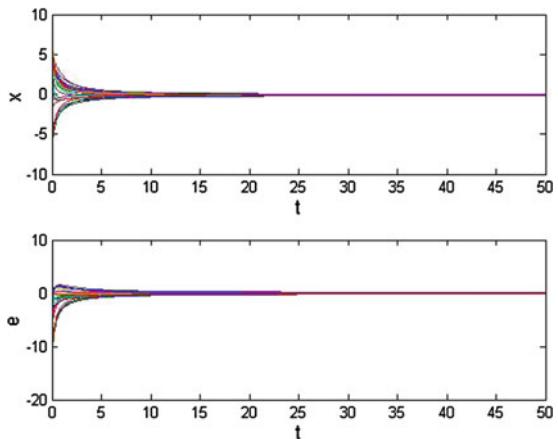
$$\mathcal{L} = \begin{pmatrix} 3 & 0 & 0 & -1 & -1 & -1 \\ -1 & 1 & 0 & 0 & 0 & 0 \\ -1 & -1 & 2 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 \end{pmatrix},$$

with its nonzero eigenvalues 1, 2, 3.3247, and $1.3376 \pm i0.5623$.

The agent dynamics and consensus protocol are given by Eqs. (9.1) and (9.3), respectively, with

$$A = \begin{pmatrix} -2 & 2 \\ -1 & 1 \end{pmatrix}, B = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, C = \begin{pmatrix} 1 & 0 \end{pmatrix},$$

Fig. 9.2 The solution trajectory and error vector of system (9.7)



$$F = \begin{pmatrix} 1 \\ -1 \end{pmatrix}, \text{ and } K = \begin{pmatrix} -1 & 2 \end{pmatrix}.$$

When $\alpha = 0.60$ and $c = 0.28$, then the eigenvalues of \mathcal{L} satisfy the condition

$$\begin{aligned} \min\{|\arg(\text{spec}(A + c\lambda_i FC))|, |\arg(\text{spec}(A + BK))|\} \\ = 2.7436 > 0.9425 = \alpha \frac{\pi}{2}. \end{aligned}$$

For convenience, we use e and x to denote respectively the consensus error vector $\delta(t)$ and the state vector $x(t)$ with $x(t) = [x_1(t)^T, \dots, x_N(t)^T]^T$. The error vector e and the solution trajectory of system (9.7) are given in Fig. 9.2. It is easy to see that system (9.7) is asymptotically stable.

9.2.3 Conclusion

This section has studied the distributed consensus problem of fractional-order multi-agent systems under a time-invariant communication topology with each agent having a general form of fractional dynamics. A fractional-order observer-type consensus protocol based on relative output measurements between neighboring agents has been proposed, where linear matrix inequality stability conditions for fractional-order systems have been used. A sufficient condition and a necessary and sufficient condition for the consensus of fractional-order multi-agent systems have been proposed and analyzed finally. It is well known that there may exist time delay in communication among agents, so the general fractional-order multi-agent systems with communication delay will be investigated in the future work.

9.3 Novel Fractional-Order Observer Design for Consensus Tracking of Multi-agent Systems with Second-Order Dynamics

In this section, fractional-order controller is designed for the followers to achieving the tracking consensus in multi-agent systems where the velocity information is unavailable. Motivated by the broad applications of fractional-order multi-agent systems, this section considers the leader-following case in directed topology where the leader is described by the second-order differential system while the followers are governed by fractional-order dynamical systems. It is found that the follower can track the dynamics of the leader only by exchanging the position information with each neighbors and the leader if this follower is informed. This result is better than the general integer leader-following case which uses both the position and velocity information of each neighbors.

The dynamics of the leader are described by the second-order system

$$\begin{cases} \dot{x}_0(t) = v_0(t), \\ \dot{v}_0(t) = u_0(t), \end{cases} \quad (9.21)$$

where $x_0(t)$ and $u_0(t)$ represent, respectively, the state vector and the control input for the leader.

The fractional-order observer of the followers is designed as follows:

$$x_i^{(\alpha)}(t) = - \sum_{j \in \mathcal{N}_i} a_{ij}(x_i(t) - x_j(t)) - b_i(x_i(t) - x_0(t)) + u_i(t), \quad i = 1, 2, \dots, N, \quad (9.22)$$

where $x_i(t) \in R^n$ is the position state of the i th follower, $a_{ij} = -L_{ij}$, b_i is the weight between the leader and the i th follower, $b_i \neq 0$ if the i th follower have access to the states of the leader directly; $b_i = 0$, otherwise, and $u_i(t)$ is the control input of the i th follower. For notational simplicity, $n = 1$ is considered throughout this section, but all the results obtained can be easily generalized to the case with $n > 1$ by using the Kronecker product operations.

To derive the main results, the following assumption is made throughout this section.

Assumption 9.1 For any fixed $u_0(t)$, there exists $u_i(t)$ for each $i \in N$ such that $s^{-2}\mathcal{L}(u_0(t)) = s^{-\alpha}\mathcal{L}(u_i(t))$. $u_0(t)$ and $u_i(t)$ are the control inputs for the leader and the i th follower, respectively.

Remark 9.3 Note that once $u_0(t)$ is given, $u_i(t)$ can be calculated by the inverse Laplace transform such that $s^{-2}\mathcal{L}(u_0(t)) = s^{-\alpha}\mathcal{L}(u_i(t))$. Thus, Assumption 9.1 is reasonable.

Definition 9.5 The leader-following systems are said to achieve consensus if for any initial conditions,

$$\lim_{t \rightarrow +\infty} \|x_i(t) - x_0(t)\| = 0, \quad \forall i = 1, 2, \dots, N.$$

9.3.1 Main Results for $\alpha \in (1, 2]$

Theorem 9.3 Suppose that the leader has a directed path to every follower and Assumption 9.1 holds. The leader-following systems (9.21) and (9.22) can reach consensus if and only if

$$\min_{1 \leq i \leq N} |\arg(\lambda_i(-(L + B)))| > \frac{\alpha\pi}{2}, \quad \alpha \in (1, 2),$$

where $\lambda_i(-(L + B))$ are the eigenvalues of $-(L + B)$.

Proof Taking the Laplace transform of system (9.21) gives

$$s^2 X_0(s) - sx_0(0) - \dot{x}_0(0) = \mathcal{L}(u_0(t)),$$

where $X_0(s) = \mathcal{L}(x_0(t))$. It is to easy see that

$$X_0(s) = s^{-1}x_0(0) + s^{-2}\dot{x}_0(0) + s^{-2}\mathcal{L}(u_0(t)). \quad (9.23)$$

Taking the Laplace transform of system (9.22) gives

$$\begin{aligned} & s^\alpha X_i(s) - s^{\alpha-1}x_i(0) - s^{\alpha-2}\dot{x}_i(0) \\ &= - \sum_{j \in \mathcal{N}_i} a_{ij}(X_i(s) - X_j(s)) - b_i(X_i(s) - X_0(s)) + \mathcal{L}(u_i(t)), \end{aligned}$$

where $X_k(s) = \mathcal{L}(x_k(t))$, $k = 1, 2, \dots, N$. It is to easy to see that

$$\begin{aligned} X_i(s) &= s^{-1}x_i(0) + s^{-2}\dot{x}_i(0) - s^{-\alpha} \sum_{j \in \mathcal{N}_i} a_{ij}(X_i(s) - X_j(s)) \\ &\quad - s^{-\alpha}b_i(X_i(s) - X_0(s)) + s^{-\alpha}\mathcal{L}(u_i(t)). \end{aligned} \quad (9.24)$$

Let $e_i(t) = x_i(t) - x_0(t)$ be the error vector. From Definition 9.5, the followers can reach consensus with the leader if the error vectors satisfy:

$$\lim_{t \rightarrow +\infty} \|e_i(t)\| = 0, \quad \forall i = 1, 2, \dots, N.$$

Taking the Laplace transform of error vector gives $E_i(s) = \mathcal{L}(e_i(t)) = X_i(s) - X_0(s)$.

Subtracting (9.23) from (9.24) gives the error systems:

$$\begin{aligned} E_i(s) = & s^{-1}e_i(0) + s^{-2}\dot{e}_i(0) - s^{-\alpha} \sum_{j \in \mathcal{N}_i} a_{ij}(E_i(s) - E_j(s)) \\ & - s^{-\alpha}b_i E_i(s). \end{aligned}$$

The above error systems can be written in the following compact matrix form:

$$E(s) = s^{-1}e(0) + s^{-2}\dot{e}(0) - s^{-\alpha}LE(s) - s^{-\alpha}BE(s), \quad (9.25)$$

where $E(s) = (E_1(s), E_2(s), \dots, E_N(s))^T$, $e(0) = (e_1(0), e_2(0), \dots, e_N(0))^T$, $\dot{e}(0) = (\dot{e}_1(0), \dot{e}_2(0), \dots, \dot{e}_N(0))^T$, and $B = \text{diag}(b_1, b_2, \dots, b_N)$ is the leader adjacency matrix.

Solving for $E(s)$ yields

$$E(s) = (s^\alpha I + L + B)^{-1}(s^{\alpha-1}e(0) + s^{\alpha-2}\dot{e}(0)). \quad (9.26)$$

Then, the leader-following tracking consensus problem of systems (9.21) and (9.22) can be transformed into the asymptotically stable problem of the error system (9.26).

The characteristic equation of system (9.26) is

$$\det(s^\alpha I + L + B) = 0, \quad (9.27)$$

which is the same as that of fractional-order system:

$$x^{(\alpha)}(t) = -(L + B)x(t), \quad 1 < \alpha < 2. \quad (9.28)$$

It is easy to see that systems (9.21) and (9.22) can reach tracking consensus if system (9.28) is asymptotically stable. So the leader-following consensus problem can be casted to the asymptotically stable problem of the zero equilibrium point of system (9.28). From Lemma 9.2, it can be seen that system (9.28) is asymptotically stable if and only if

$$\min_{1 \leq i \leq N} |\arg(\lambda_i(-(L + B)))| > \frac{\alpha\pi}{2}, \quad \alpha \in (1, 2).$$

This completes the proof.

Remark 9.4 It is required that the augmented digraph formed by the leader and the followers should contain a directed spanning tree such that the leader can directly or indirectly affect all the followers, which means that there is at least one entry of $B = \text{diag}(d_1, d_2, \dots, d_N)$ being positive. Otherwise, there must exist a follower

which is not influenced by the leader. Then, that follower can not track the leader and the whole multi-agent system can not agree on a homogeneous consensus state. In addition, if there is at least one entry of B being positive, then $L + B$ is positive definite from [10].

Remark 9.5 The condition in Theorem 9.3 means that the roots of characteristic equation of $-(L + B)$ should all locate in left-open complex plane.

Let $\mu_i, i = 1, 2, \dots, N$, be the eigenvalues of $-(L + B)$. Note that $\det(s^\alpha I + L + B) = \prod_{i=1}^N (s^\alpha - \mu_i)$. It can be found that the real and imaginary parts of μ_i satisfy some relationship expressions since $s^\alpha - \mu_i = 0, i = 1, 2, \dots, N$.

Corollary 9.1 Suppose that the augmented digraph formed by the leader and all the followers contains a directed spanning tree and Assumption 9.1 holds. Then, the leader-following systems (9.21) and (9.22) can reach consensus if and only if

$$\tan\left(\frac{\pi}{2}(\alpha - 2)\right) < \frac{\operatorname{Im}(\mu_i)}{\operatorname{Re}(\mu_i)} < \tan\left(\frac{\pi}{2}(2 - \alpha)\right),$$

where $\mu_i, i = 1, 2, \dots, N$ are the eigenvalues of $-(L + B)$, $\operatorname{Re}(\mu_i)$ and $\operatorname{Im}(\mu_i)$ represent the real and imaginary parts of μ_i , respectively.

Proof Let $\mu_i = a_i + jb_i$, where j is the imaginary unit. a_i and b_i represent the real and imaginary parts of μ_i , respectively. If the leader has a directed path to every follower, the real parts of eigenvalues of matrix $L + B$ are negative, that is, $a_i < 0$. The argument principal value of μ_i is then discussed below under two cases: $b_i > 0$ and $b_i < 0$. Suppose that $\arg(\cdot)$ denotes the argument principle of a complex number, which takes value in $(-\pi, \pi]$. Then, one has

$$\begin{cases} \arg(\mu_i) = \pi + \arctan \frac{b_i}{a_i}, & b_i > 0, \\ \arg(\mu_i) = \arctan \frac{b_i}{a_i} - \pi, & b_i < 0. \end{cases}$$

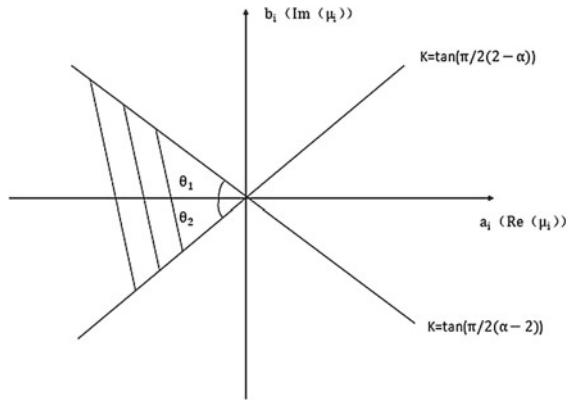
Based on Theorem 9.3, the followers can track the states of the leader if and only if $\min_{1 \leq i \leq N} |\arg(\mu_i)| > \frac{\pi\alpha}{2}$. That is

$$\begin{cases} \pi + \arctan \frac{b_i}{a_i} > \frac{\pi\alpha}{2}, & b_i > 0, \\ \pi - \arctan \frac{b_i}{a_i} > \frac{\pi\alpha}{2}, & b_i < 0. \end{cases}$$

i.e.,

$$\begin{cases} \frac{b_i}{a_i} > \tan\left(\frac{\pi\alpha}{2} - \pi\right), & b_i > 0, \\ \frac{b_i}{a_i} < \tan\left(\frac{\pi - \pi\alpha}{2}\right), & b_i < 0. \end{cases}$$

Fig. 9.3 The range of a_i and b_i



Since $a_i < 0$, we get

$$a_i \tan\left(\frac{\pi}{2}(2-\alpha)\right) < b_i < a_i \tan\left(\frac{\pi}{2}(\alpha-2)\right).$$

Note that $\alpha \in (1, 2]$, then the ranges of a_i and b_i is fixed once the value of α is given, see the following remark for more details:

Remark 9.6 Note that when $b_i > 0$, the larger $\arctan(b_i/a_i)$ is, the easier the inequality $\pi + \arctan(b_i/a_i) > (\pi\alpha/2)$ can satisfy. That is, the angle θ_1 in Fig. 9.3 is the bigger the better, which is the same as angle θ_2 by symmetry. This indicates that the followers can track the dynamics of the leader more easily if the shadow region showed in Fig. 9.3 is more wider.

Remark 9.7 In both Theorem 9.3 and Corollary 9.1, the followers cannot track the leader if $\alpha = 2$, since $\alpha = 2$ does not satisfy the condition in Theorem 9.3. In addition, the followers are also described by second-order system if $\alpha = 2$. It is well known that, these followers cannot achieve tracking consensus only by using current position information of neighbors' from previous studies [11].

9.3.2 Case with Identical Communication Delay

It is well known that the communication delay cannot be avoided in the practical applications due to the constraint of the equipments. So in this section, the case with identical communication delay will be discussed.

The fractional-order dynamics of the followers are governed by the following system:

$$x_i^{(\alpha)}(t) = - \sum_{j \in \mathcal{N}_i} a_{ij}(x_i(t - \tau) - x_j(t - \tau)) - b_i(x_i(t - \tau) - x_0(t - \tau)) + u_i(t), \quad (9.29)$$

where $x_i(t)$ is the position state of the i th follower, $a_{ij} = -L_{ij}$, b_i is the weight between the leader and the i th follower, $b_i \neq 0$ if the i th follower can get information from leader directly, $b_i = 0$, otherwise, τ is the communication delay, and $u_i(t)$ is the control input of the i th follower.

In the following, a necessary and sufficient condition will be derived to ensure the leader-following consensus of multi-agent systems (9.21) and (9.29), where the bound of τ is explicitly analyzed by using the generalized Nyquist stability criterion.

Theorem 9.4 Suppose that the leader has a directed path to every follower and Assumption 9.1 holds. The leader-following systems (9.21) and (9.29) can reach consensus if and only if

$$\tau < \min_{1 \leq i \leq N} \frac{\pi + \arg(\mu_i) - \frac{\pi\alpha}{2}}{|\mu_i|^{\frac{1}{\alpha}}}, \quad (9.30)$$

where μ_i is the i th eigenvalue of $L + B$, B is the leader adjacency matrix, and $\arg(\cdot)$ denotes the argument of a complex number.

Proof When $\alpha \in (1, 2]$, taking the Laplace transform of system (9.29) gives

$$\begin{aligned} & s^\alpha X_i(s) - s^{\alpha-1} x_i(0) - s^{\alpha-2} \dot{x}_i(0) \\ &= - \sum_{j \in \mathcal{N}_i} a_{ij} e^{-\tau s} (X_i(s) - X_j(s)) - b_i e^{-\tau s} (X_i(s) - X_0(s)) + \mathcal{L}(u_i(t)), \end{aligned} \quad (9.31)$$

where $X_k(s) = \mathcal{L}(x_k(t))$, $k = 1, \dots, N$. By simple calculation, Eq. (9.31) can be written as

$$\begin{aligned} X_i(s) &= s^{-1} x_i(0) + s^{-2} \dot{x}_i(0) - s^{-\alpha} \sum_{j \in \mathcal{N}_i} a_{ij} e^{-\tau s} (X_i(s) - X_j(s)) \\ &\quad - s^{-\alpha} b_i e^{-\tau s} (X_i(s) - X_0(s)) + s^{-\alpha} \mathcal{L}(u_i(t)). \end{aligned} \quad (9.32)$$

Let $e_i(t) = x_i(t) - x_0(t)$, subtracting (9.23) from (9.32) gives the error system

$$\begin{aligned} E_i(s) &= s^{-1} e_i(0) + s^{-2} \dot{e}_i(0) - s^{-\alpha} \sum_{j \in \mathcal{N}_i} a_{ij} e^{-\tau s} (E_i(s) - E_j(s)) \\ &\quad - s^{-\alpha} b_i e^{-\tau s} (E_i(s) - E_0(s)), \end{aligned}$$

which can be written in the following compact matrix form

$$E(s) = s^{-1} e(0) + s^{-2} \dot{e}(0) - s^{-\alpha} L e^{-\tau s} E(s) - s^{-\alpha} B e^{-\tau s} E(s), \quad (9.33)$$

where $E(s) = (E_1(s), E_2(s), \dots, E_N(s))^T$, $e(0) = (e_1(0), e_2(0), \dots, e_N(0))^T$, $\dot{e}(0) = (\dot{e}_1(0), \dot{e}_2(0), \dots, \dot{e}_N(0))^T$, and $B = \text{diag}(b_1, b_2, \dots, b_N)$ is the leader adjacency matrix.

Solving for $E(s)$ yields

$$E(s) = (s^\alpha I + (L + B)e^{-\tau s})^{-1}(s^{\alpha-1}e(0) + s^{\alpha-2}\dot{e}(0)). \quad (9.34)$$

The characteristic equation is

$$\det(s^\alpha I + (L + B)e^{-\tau s}) = 0. \quad (9.35)$$

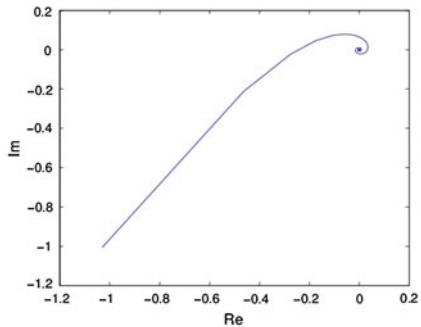
Let δ_i be the i th eigenvalue of $L + B$. Note that $\det(s^\alpha I + (L + B)e^{-\tau s}) = \prod_{i=1}^N (s^\alpha + \delta_i e^{-\tau s})$. According to the generalized Nyquist stability criterion, all the roots of $s^\alpha + \delta_i e^{-\tau s} = 0$ are on the open left half plane if and only if the Nyquist plot of $G_i(\omega) = \delta_i e^{-j\omega\tau} / (\omega^\alpha)$ neither encircle nor touch the point $(-1, j0)$ for all $\omega \in (-\infty, +\infty)$. Here only the case when $\omega \in (0, +\infty)$ is considered due to the symmetric property of the Nyquist plot. The bound on the identical communication delay τ can be calculated as follows

$$\begin{aligned} G_i(\omega) &= \frac{|\delta_i| e^{j\arg(\delta_i)e^{-j\omega\tau}}}{\omega^\alpha e^{\frac{j\pi\alpha}{2}}} \\ &= \frac{|\delta_i|}{\omega^\alpha} e^{j(\arg(\delta_i) - \omega\tau - \frac{\pi\alpha}{2})} \\ &= \frac{|\delta_i|}{\omega^\alpha} \left(\cos \left(\omega\tau + \frac{\pi\alpha}{2} - \arg(\delta_i) \right) - j \sin \left(\omega\tau + \frac{\pi\alpha}{2} - \arg(\delta_i) \right) \right). \end{aligned}$$

The Nyquist plot of $G_i(\omega)$ is illustrated in Fig. 9.4. It is easy to see that the Nyquist plot of $G_i(\omega)$ does not enclose $(-1, j0)$ if and only if the intersection point where $G_i(\omega)$ intersects with the real axis for the first time when ω evolves from 0 to $+\infty$ is on the right side of $(-1, j0)$, which is equivalent to $\omega\tau + (\pi\alpha/2) - \arg(\delta_i) = \pi$, and $|\delta_i|/\omega^\alpha < 1$. This implies $\tau < \min_{1 \leq i \leq N} (\pi - (\pi\alpha/2) + \arg(\delta_i)) / |\delta_i|^{\frac{1}{\alpha}}$. Therefore, under condition (1.30), the zero equilibrium point of leader-following error system is asymptotically stable.

Remark 9.8 Note that, in condition (9.30), $|\delta_i|$ and $\arg(\delta_i)$ are all determined by the real and imaginary parts of δ_i , $i = 1, 2, \dots, N$, which means that Theorem 9.4 establishes the relationship between the identical communication delay τ and the real and imaginary parts of eigenvalues of the Laplacian matrix L and the leader adjacency matrix B . So the real and imaginary parts of δ_i play important roles in the threshold of τ .

Fig. 9.4 Nyquist plot of $G_i(j\omega)$ for
 $\delta_i = 2.5 + 0.86601j$,
 $\alpha = 1.5$, and $\tau = 0.23$



9.3.3 Numerical Simulation

In this section, some examples are given to verify the effectiveness of the theoretical analysis.

Example 9.1 Assume that the communication topology is shown in Fig. 9.5, where node 0 is the leader while nodes 1, 2, 3, and 4 are the followers. The corresponding Laplacian matrix is

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

and the leader adjacency matrix is

Fig. 9.5 The communication topology

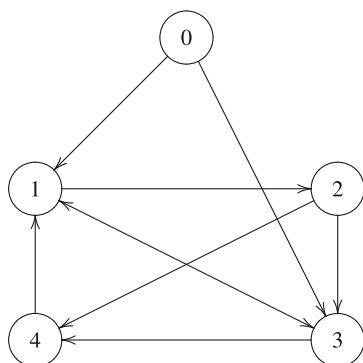


Fig. 9.6 The solution trajectories of the leader and the followers

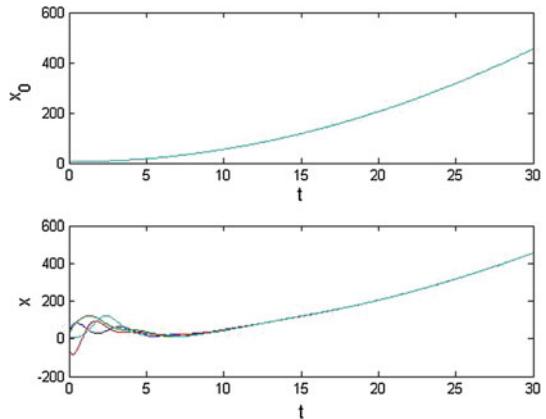
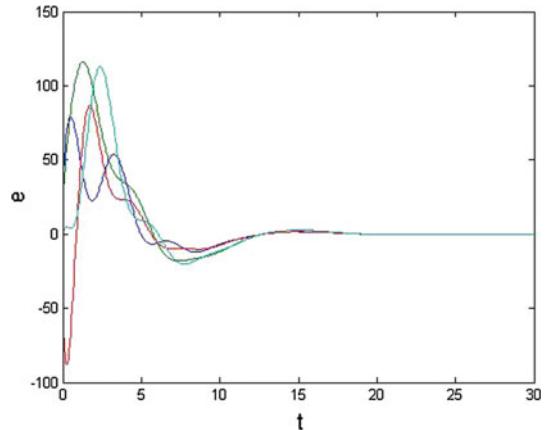


Fig. 9.7 The trajectories of the error system



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

Some calculations give that the eigenvalues of $L + B$ are 0.3820, 2.6180, and $2.5000 \pm 0.86601i$.

Choose $\alpha = 1.50$, then the $\min_{1 \leq i \leq 4} |\arg(\lambda_i(-(L + B)))| = 2.8081 > 2.6180 = \alpha\pi/2$. By using Theorem 9.3, the leader-following consensus can be ensured. For the brevity of expression, we use x_0 , x , and e to denote $x_0(t)$, $x(t)$ with $x(t) = [x_1(t), x_2(t), \dots, x_N(t)]$, and $e(t)$ with $e(t) = [e_1(t), e_2(t), \dots, e_N(t)]$, respectively. Then, one can see that the followers track the leader asymptotically as shown in Figs. 9.6 and 9.7.

Example 9.2 In the presence of identical communication delay, choose the communication topology the same as that in Example 9.1. By checking condition (9.30) in Theorem 9.4, one has that the leader-following consensus can be achieved if $\tau < 0.2363$. Taking $\tau = 0.23$, the trajectory of the leader and the followers as well as the error are shown in Figs. 9.8 and 9.9. One can see that the systems can achieve tracking consensus. However, if the communication delay τ takes value of 0.31, which exceeds the threshold, it can be found that the followers can not track the leader as shown in Figs. 9.10 and 9.11.

Fig. 9.8 The solution trajectories of the leader and the followers when $\tau = 0.23$

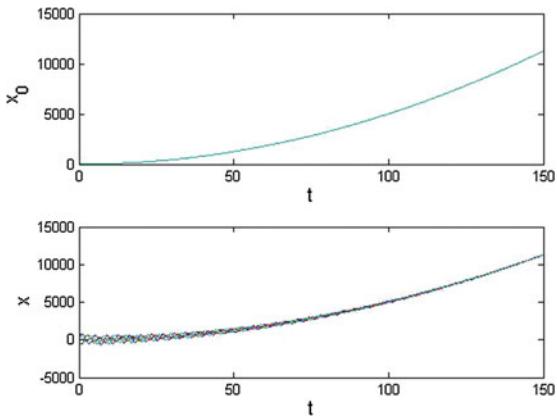


Fig. 9.9 The trajectories of the error system when $\tau = 0.23$

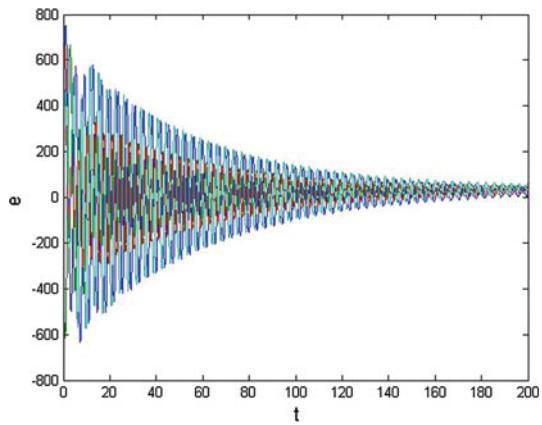


Fig. 9.10 The solution trajectories of the leader and the followers when $\tau = 0.31$

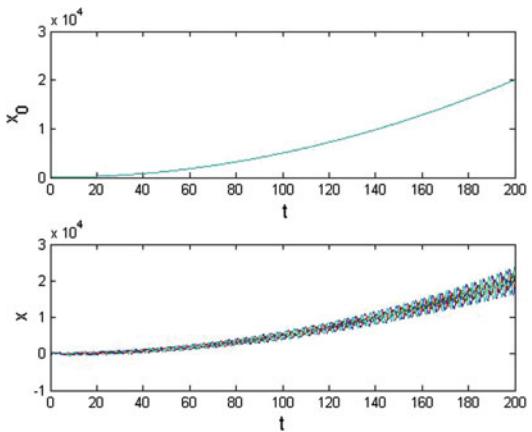
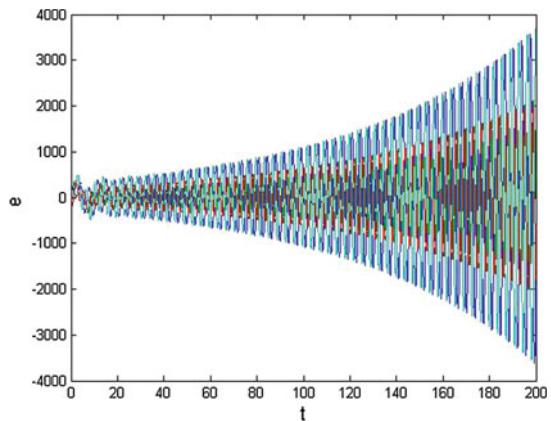


Fig. 9.11 The trajectories of the error system when $\tau = 0.31$



9.3.4 Conclusion

In this section, the leader-following consensus has been studied in multi-agent systems in which the leader is described by the second-order differential system while the followers are governed by fractional-order differential systems. First, the case without communication delay has been considered. It has been found that the followers can track the dynamics of the leader only by using the local position information of the neighbors if some followers are informed when $\alpha \in (1, 2)$. This result is better than the previous studies about the second-order leader-following consensus which uses both position and velocity information of neighbors, and this is also the main contribution of this section. For this model, a necessary and sufficient condition for the leader-following consensus has been proposed. Then, the case with identical communication delay has been investigated. It has been found that, in both cases, the real and imaginary parts of the eigenvalues of the Laplacian matrix of the

topology graph and the leader's adjacency matrix play important roles in the tracking consensus condition and the threshold of allowable identical communication delay.

9.4 Stabilization Consensus of Fractional-Order Multi-agent Systems with Uncertain Parameters

In this section, the stabilization consensus problem of fractional-order multi-agent systems with uncertain parameters is discussed via designing the decentralized controllers based on relative information of neighbors and absolute information of some agents. Before studying this problem, some lemmas are needed.

Lemma 9.6 [20, 21] Let $A \in R^{n \times n}$, $0 < \alpha < 1$ and $\theta = (1-\alpha)\pi/2$. The fractional-order system $d^\alpha x(t)/dt^\alpha = Ax(t)$ is asymptotically stable if and only if there exists a positive definite Hermitian matrix $X = X^* > 0$, $X \in C^{n \times n}$ such that

$$(rX + \bar{r}\bar{X})^T A^T + A(rX + \bar{r}\bar{X}) < 0,$$

where $r = e^{\theta i}$. Moreover, the matrix $(rX + \bar{r}\bar{X})$ in the above inequality is real and nonsingular.

Lemma 9.7 [20, 21] Let $A \in R^{n \times n}$, $1 \leq \alpha < 2$ and $\theta = (1-\alpha)\pi/2$. The fractional-order system $d^\alpha x(t)/dt^\alpha = Ax(t)$ is asymptotically stable if and only if there exists a positive definite symmetric matrix $X = X^T > 0$, $X \in R^{n \times n}$ such that

$$\begin{bmatrix} (A^T X + XA)\sin\theta & (XA - A^T X)\cos\theta \\ \bullet & (A^T X + XA)\sin\theta \end{bmatrix} < 0,$$

where the symbol ‘•’ represents the symmetric term in a symmetric matrix.

Remark 9.9 Define

$$\Theta = \begin{bmatrix} \sin\theta & -\cos\theta \\ \cos\theta & \sin\theta \end{bmatrix},$$

then the condition (1.37) can be rewritten as

$$\text{Sym}\{\Theta \otimes (A^T X)\} < 0,$$

which is equivalent to

$$\text{Sym}\{\Theta \otimes (AX)\} < 0.$$

since A is similar to A^T , where $\text{Sym}\{A\} = A + A^T$.

Before ending this section, the following lemma is provided which can be found in many text books on matrix theory, e.g., [27].

Lemma 9.8 Let X and Y be real matrices with appropriate dimensions. Then, for any positive definite symmetric matrix with appropriate dimensions $\Lambda > 0$, we have

$$X^T Y + Y^T X \leq X^T \Lambda X + Y^T \Lambda^{-1} Y.$$

9.4.1 Main Results

In this section, the problem formulation is first provided. Then, the main theoretical results of this section are presented and analyzed.

Consider the following fractional-order linear time-invariant multi-agent system:

$$D^\alpha x_i(t) = (A + \Delta A_i(t))x_i(t) + Bu_i(t), \quad i = 1, 2, \dots, N, \quad (9.36)$$

where $\alpha \in (0, 2)$ is the fractional commensurate order, $x_i(t) \in R^n$ and $u(t) \in R^l$ are, respectively, the state vector and control input of the i th agent. $A \in R^{n \times n}$ and $B \in R^{n \times l}$ are known real constant matrices, $\Delta A_i(t) \in R^{n \times n}$ is time-variant matrix representing the parameter uncertainty satisfying the following condition:

$$\Delta A_i(t) = DF_i(t)E, \quad (9.37)$$

and D, E are some constant matrices with appropriate dimensions which characterise the structure of the uncertainty, $F_i(t)$ is the uncertainty satisfying $F_i^T(t)F_i^T(t) \leq \delta^2 I$, in which $I \in R^{n \times n}$ denotes the identity matrix, and $\delta > 0$ is a given constant.

The communication topology of system (9.36) is represented by a directed graph \mathcal{G} . It is assumed that only a part of agents can measure their own states while each agent can measure the relative states with respect to its neighbours.

Assumption 9.2 The directed communication graph \mathcal{G} is strongly connected and at least one agent knows its own state.

Based on the absolute states of a portion of agents and the relative states between neighbouring agents, a decentralized controller is designed as follows:

$$u_i(t) = cK(\sum_{j=1}^N a_{ij}(x_i(t) - x_j(t)) + d_i x_i(t)), \quad i = 1, 2, \dots, N, \quad (9.38)$$

where $c > 0$ is a constant which denotes the coupling strength, $K \in R^{l \times n}$ is the feedback gain matrix to be designed, $a_{ij} = -L_{ij}$, and d_i are constant scalars: $d_i > 0$ if the i -th agent knows its own state information; $d_i = 0$ otherwise.

Let $x(t) = [x_1^T(t), x_2^T(t), \dots, x_N^T(t)]^T$ and $\tilde{D} = \text{diag}(d_1, d_2, \dots, d_N)$. Under controller (9.38), the multi-agent system (9.36) can be rewritten as:

$$x^\alpha(t) = (I_N \otimes A + c\hat{\mathcal{L}} \otimes BK + (I_N \otimes D)\Delta(I_N \otimes E))x(t), \quad (9.39)$$

where $\hat{\mathcal{L}} = \mathcal{L} + \tilde{D}$, \mathcal{L} is the Laplace matrix of \mathcal{G} and $\Delta = \text{diag}(F_1, F_2, \dots, F_N)$.

Definition 9.6 The system $D^\alpha x(t) = Ax(t) + Bu(t)$ is said to be stabilizable if there exists a linear state feedback stabilizing controller $u(t) = Kx(t)$, $K \in R^{l \times n}$ such that the resultant closed-loop system $D^\alpha x(t) = (A+BK)x(t)$ is asymptotically stable.

Definition 9.7 The perturbed fractional-order system (9.36) is said to be robustly stabilizable if there exists a linear state feedback stabilizing controller $u(t) = Kx(t)$, $K \in R^{l \times n}$ such that the resultant closed-loop system

$$D^\alpha x_i(t) = (A + \Delta A_i + BK)x_i(t), \quad (9.40)$$

is robustly asymptotically stable. In this case, $u_i(t) = Kx_i(t)$ is a robust linear state feedback stabilizing controller for system (9.36).

Remark 9.10 The fractional-order system is robustly stabilizable only if the nominal fractional-order system $D^\alpha x(t) = Ax(t) + Bu(t)$ is stabilizable. When the nominal fractional-order system is not stabilizable, the perturbed fractional-order system will never be robust stabilizable.

Theorem 9.5 Suppose $0 < \alpha < 1$ and the nominal fractional-order system is stabilizable. The perturbed fractional-order system (9.36) with controller (9.38) is robustly stabilizable if there exist a positive definite Hermitian matrix $X = X^* \in C^{n \times n} > 0$, a matrix $Y \in R^{l \times n}$ and a real constant $\varepsilon > 0$ such that

$$\Omega = \begin{bmatrix} \Omega_{11} & \bullet & \bullet \\ I_N \otimes E(rX + \bar{r}\bar{X}) & -\varepsilon^{-1}\Psi & 0 \\ I_N \otimes D^T & 0 & -\frac{\varepsilon}{\delta^2}\Psi \end{bmatrix} < 0, \quad (9.41)$$

where $\Omega_{11} = \text{Sym}I_N \otimes (A(rX + \bar{r}\bar{X})) + c\hat{\mathcal{L}} \otimes BY$, $\Psi = I_N \otimes I_N$. Then, there exists a linear state feedback stabilizing controller $u_i(t) = Kx_i(t)$ with $K = Y(rX + \bar{r}\bar{X})^{-1}$.

Proof By the Schur complement lemma, the inequality (9.41) is equivalent to

$$\begin{aligned} & \text{Sym}\{I_N \otimes (A(rX + \bar{r}\bar{X})) + c\hat{\mathcal{L}} \otimes BY\} + \varepsilon I_N \otimes (rX + \bar{r}\bar{X})^T E^T E(rX + \bar{r}\bar{X}) \\ & + \varepsilon^{-1}\delta^2(I_N \otimes DD^T) < 0. \end{aligned} \quad (9.42)$$

It follows from Lemma 9.8 that

$$\begin{aligned}
& \text{Sym}\{(I_N \otimes D)\Delta(I_N \otimes E(rX + \bar{r}\bar{X}))\} \\
&= (I_N \otimes (rX + \bar{r}\bar{X})^T E^T)\Delta^T(I_N \otimes D^T) + (I_N \otimes D)\Delta(I_N \otimes E(rX + \bar{r}\bar{X})) \\
&\leq (I_N \otimes (rX + \bar{r}\bar{X})^T E^T)(I_N \otimes \Lambda)(I_N \otimes E(rX + \bar{r}\bar{X})) \\
&\quad + ((I_N \otimes D)\Delta)(I_N \otimes \Lambda^{-1})(\Delta^T(I_N \otimes D^T)) \\
&\leq \varepsilon I_N \otimes (rX + \bar{r}\bar{X})^T E^T E(rX + \bar{r}\bar{X}) + \varepsilon^{-1}\delta^2(I_N \otimes DD^T),
\end{aligned} \tag{9.43}$$

where $\Lambda = \varepsilon I_N$. Then, from (9.42) and (9.43), we have

$$\text{Sym}\{I_N \otimes (A(rX + \bar{r}\bar{X})) + c\hat{\mathcal{L}} \otimes BY + (I_N \otimes D)\Delta(I_N \otimes E(rX + \bar{r}\bar{X}))\} < 0, \tag{9.44}$$

Let $K = Y(rX + \bar{r}\bar{X})^{-1}$, then (9.44) can be rewritten as

$$\begin{aligned}
& \text{Sym}\{(I_N \otimes A + c\hat{\mathcal{L}} \otimes BK \\
&\quad + (I_N \otimes D)\Delta(I_N \otimes E))(I_N \otimes (rX + \bar{r}\bar{X}))\} < 0.
\end{aligned} \tag{9.45}$$

It follows from (9.45) that $\text{Sym}\{((A + \Delta A_i) + BK)(rX + \bar{r}\bar{X})\} < 0$ holds for each system $x_i^\alpha(t) = (A + \Delta A_i)x_i(t) + Bu_i(t)$. From Lemma 9.6, one gets that the perturbed fractional-order system (9.36) is robustly stabilizable under controller (9.38). This completes the proof.

Theorem 9.5 gives a sufficient condition for system (9.36) to reach stabilization consensus under controller (9.38) when the fractional order belongs into $(0, 1)$. The condition for $\alpha \in [1, 2]$ will be given in the following theorem.

Theorem 9.6 Suppose $1 \leq \alpha < 2$ and the nominal fractional-order system is stabilizable. The perturbed fractional-order system (9.36) with controller (9.38) is robustly stabilizable if there exist a positive definite symmetrix matrix $X = X^T \in R^{n \times n} > 0$, a matrix $Y \in R^{l \times n}$ and a real constant $\varepsilon > 0$ such that

$$\mathcal{Q} = \begin{bmatrix} \Xi_{11} & \bullet & \bullet \\ I_N \otimes EX & -\varepsilon^{-1}\Psi & 0 \\ I_N \otimes D^T \Theta^T & 0 & -\frac{\varepsilon}{\delta^2}\Psi \end{bmatrix} < 0, \tag{9.46}$$

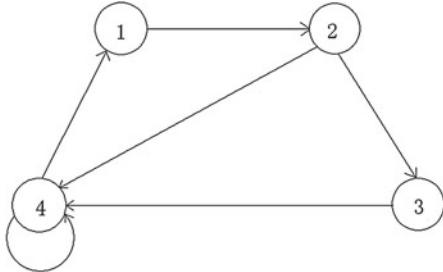
where $\Xi_{11} = \text{Sym}\{(I_N \otimes \Theta)(I_N \otimes AX + c\hat{\mathcal{L}} \otimes BY)\}$ and $\Psi = I_N \otimes I_N$. Then, there exists a linear state feedback stabilizing controller $u_i(t) = Kx_i(t)$ with $K = YX^{-1}$.

Proof By the Schur complement lemma, the inequality (9.46) is equivalent to

$$\begin{aligned}
& \text{Sym}\{(I_N \otimes D)(I_N \otimes AX + c\hat{\mathcal{L}} \otimes BY)\} \\
&+ \varepsilon(I_N \otimes X^T E^T EX) + \varepsilon^{-1}\delta^2(I_N \otimes \Theta DD^T \Theta) < 0.
\end{aligned} \tag{9.47}$$

It follows from Lemma 9.7 that

Fig. 9.12 The Communication Topology



$$\begin{aligned}
 & Sym\{(I_N \otimes \Theta)(I_N \otimes D)\Delta(I_N \otimes E)(I_N \otimes X)\} \\
 & = Sym\{(I_N \otimes \Theta D)\Delta(I_N \otimes EX)\} \\
 & = (I_N \otimes X^T E^T)\Delta^T(I_N \otimes D^T \Theta^T) + (I_N \otimes \Theta D)\Delta(I_N \otimes EX) \\
 & \leq (I_N \otimes X^T E^T)(I_N \otimes \Lambda)(I_N \otimes EX) + (I_N \otimes \Theta D)\Delta(I_N \otimes \Lambda^{-1})\Delta^T(I_N \otimes D^T \Theta^T) \\
 & \leq \varepsilon(I_N \otimes X^T E^T EX) + \varepsilon^{-1}\delta^2(I_N \otimes \Theta DD^T \Theta^T), \tag{9.48}
 \end{aligned}$$

where $\Lambda = \varepsilon I_N$. Then, from (9.47) and (9.48), we have

$$\begin{aligned}
 & Sym\{(I_N \otimes D)(I_N \otimes AX + c\hat{\mathcal{L}} \otimes BY) \\
 & \quad + (I_N \otimes \Theta)(I_N \otimes D)\Delta(I_N \otimes E)(I_N \otimes X)\} < 0. \tag{9.49}
 \end{aligned}$$

Let $K = YX^{-1}$, then (9.49) can be rewritten as

$$\begin{aligned}
 & Sym\{(I_N \otimes D)(I_N \otimes A + c\hat{\mathcal{L}} \otimes BK \\
 & \quad + (I_N \otimes \Theta)(I_N \otimes D)\Delta(I_N \otimes E)(I_N \otimes X)\} < 0. \tag{9.50}
 \end{aligned}$$

It follows from (9.50) that the $Sym\{\Theta \otimes (A + \Delta A_i(t) + BK)X\} < 0$ holds for each system $x_i^\alpha(t) = (A + \Delta A_i(t))x_i(t) + Bu_i(t)$. From Lemma 9.7, each fractional-order system is robustly asymptotically stable, so the perturbed fractional-order system (9.36) is robustly stabilizable under controller (9.38). This completes the proof.

9.4.2 Numerical Simulation

Consider the uncertain fractional-order multi-agent system consisting of four agents, where the communication topology is shown in Fig. 9.12.

The corresponding Laplace matrix is

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

and the matrix \tilde{D} is

$$\tilde{D} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Thus, the matrix $\hat{L} = L + \tilde{D}$ is

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

The parameter matrices are given as follows

$$A = \begin{pmatrix} 0.4 & 0.35 \\ -0.9 & -2.7 \end{pmatrix}, \quad B = \begin{pmatrix} -1.45 \\ -0.75 \end{pmatrix},$$

$$E = \begin{pmatrix} -0.3 & 0 \\ 0 & -1 \end{pmatrix}, \quad D = \begin{pmatrix} -0.01 & -0.05 \\ -0.15 & -0.08 \end{pmatrix},$$

and

$$F = \begin{pmatrix} \sin(0.5\pi t) & 0 \\ 0 & \cos(0.5\pi t) \end{pmatrix}.$$

1. Case I when $0 < \alpha < 1$

Choose $\alpha = 0.6$. Based on Theorem 9.5, the feasible solution of LMI (9.41) by using Matlab is given as

$$rX + \bar{r}\bar{X} = \begin{pmatrix} 0.2340 & -0.1127 \\ -0.1127 & 0.3788 \end{pmatrix},$$

$$Y = (0.7045 \quad -0.1558).$$

So, the corresponding feedback gain matrix is

$$K = Y(rX + \bar{r}\bar{X})^{-1} = (3.2829 \quad 0.5651).$$

The state trajectories of the uncertain fractional-order multi-agent system consisting of four agents are shown in Fig. 9.13. It can be seen that the uncertain fractional-order multi-agent system is robustly stabilizable under control protocol (9.38) when $0 < \alpha < 1$, which verify the effectiveness of Theorem 9.5.

2. Case when $1 \leq \alpha < 2$

Suppose that $\alpha = 1.5$. Based on Theorem 9.6, the feasible solution of LMI inequality (9.46) by using Matlab is

$$X = \begin{pmatrix} 0.2670 & -0.2172 \\ -0.2172 & 0.3433 \end{pmatrix},$$

$$Y = (0.4749 \quad 0.4304).$$

So, the corresponding feedback gain matrix is

$$K = YX^{-1} = (5.7691 \quad 4.9045).$$

The state trajectory of the uncertain fractional-order multi-agent system consisting of four agents is shown in Fig. 9.14. It can be seen that the uncertain fractional-order multi-agent system is robustly stabilizable under control protocol (9.38) when $1 \leq \alpha < 2$, which verify the validity of Theorem 9.6.

9.4.3 Conclusion

The stabilization consensus problem of fractional-order multi-agent systems with uncertain parameters has been discussed under directed topologies with some self-loops. A fractional-order control protocol based on the relative information of neigh-

Fig. 9.13 The state trajectories of $x_i(t)$ ($i = 1, 2, 3, 4$) when $\alpha = 0.6$

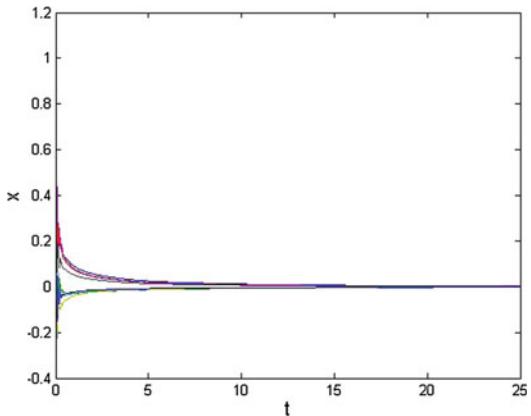
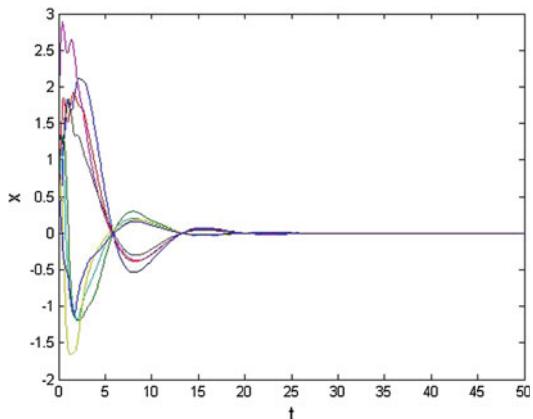


Fig. 9.14 The state trajectory of $x_i(t)(i = 1, 2, 3, 4)$ when $\alpha = 1.5$



bors as well as its absolute information has been designed. The consensus problem when $0 \leq \alpha < 1$ and $1 \leq \alpha < 2$ have been discussed respectively under the designed control protocols. Two sufficient conditions have been provided in terms of LMI. However, the conditions in these two theorems all contain the Laplace matrix which is the global information of the system. This is unpractical when the scale of the system is very large. Some improved conditions are needed to be further studied.

Acknowledgments This work was supported by the National Science Fund for Excellent Young Scholars under Grant No. 61322302, the National Natural Science Foundation of China under Grant No. 61304168, the Six Talent Peaks of Jiangsu Province of China under Grant No. 2014-DZXX-004, and the Fundamental Research Funds for the Central Universities of China.

References

1. Yu, W., Chen, G., Wang, Z., Yang, W.: Distributed consensus filtering in sensor networks. *IEEE Trans. Syst. Man Cybern.-Part B* **39**(6), 1568–1577 (2009)
2. Haghichi, R., Cheah, C.: Multi-group coordination control for robot swarms. *Automatica* **48**(10), 2526–2534 (2012)
3. Reynolds, C.: Flocks, herds and schools: a distributed behavioral model. *Comput. Graph.* **21**(4), 25–34 (1987)
4. Vicsek, T., Czirk, A., Ben-Jacob, E., Cohen, I., Shochet, O.: Novel type of phase transition in a system of self-driven particles. *Phys. Rev. Lett.* **75**(6), 1226–1229 (1995)
5. Li, Z., Duan, Z., Chen, G., Huang, L.: Consensus of multiagent systems and synchronization of complex networks: a unified viewpoint. *IEEE Trans. Circuits Syst. I: Regul. Pap.* **57**(1), 213–224 (2010)
6. Wen, G., Duan, Z., Li, Z., Chen, G.: Consensus tracking of nonlinear multi-agent systems with switching directed topologies. In: The 12th International Conference on Control Automation Robotics and Vision, pp. 889–894 (2012)
7. Zhao, Y., Duan, Z., Chen, G.: Finite-time consensus tracking for multi-agent systems with settling time estimation. In: The 33rd Chinese Control Conference, pp. 1132–1137 (2014)

8. Wen, G., Duan, Z., Chen, G., Yu, W.: Consensus tracking of multi-agent systems with Lipschitz-type node dynamics and switching topologies. *IEEE Trans. Circuits Syst. I: Regul. Pap.* **61**(2), 499–511 (2014)
9. Li, Z., Duan, Z., Xie, L., Liu, X.: Distributed robust control of linear multi-agent systems with parameter uncertainties. *Int. J. Control.* **85**(8), 1039–1050 (2012)
10. Hong, Y., Hu, J., Gao, L.: Tracking control for multi-agent consensus with an active leader and variable topology. *Automatica* **42**(7), 1177–1182 (2006)
11. Ren, W., Beard, R.: On consensus algorithms for double-integrator dynamics. *IEEE Trans. Autom. Control* **53**(6), 77–104 (2008)
12. Ren, W., Atkins, E.: Distributed multi-vehicle coordinated control via local information exchange. *Int. J. Robust Nonlinear Control* **17**(10–11), 1002–1033 (2007)
13. Hu, G.: Robust consensus tracking of a class of second-order multi-agent dynamic systems. In: *IEEE Conference on Decision and Control*, pp. 3214–3220 (2010)
14. Yu, W., Chen, G., Cao, M.: Some necessary and sufficient conditions for second-order consensus in multi-agent dynamical systems. *Automatica* **46**(6), 1089–1095 (2010)
15. Yu, W., Chen, G., Cao, M., Kurths, J.: Second-order consensus for multi-agent systems with directed topologies and nonlinear dynamics. *IEEE Trans. Syst. Man Cybern.-Part B* **40**(3), 881–891 (2010)
16. Bagley, R., Torvik, P.: On the fractional calculus model of viscoelastic behavior. *J. Rheol.* **30**(1), 133–155 (1986)
17. Cao, Y., Li, Y., Ren, W., Chen, Y.: Distributed coordination of networked fractional-order systems. *IEEE Trans. Syst. Man Cybern.-Part B* **40**(2), 362–370 (2010)
18. Cao, Y., Ren, W.: Distributed formation control for fractional-order systems: dynamic interaction and absolute/relative damping. *Syst. Control Lett.* **59**(3–4), 233–240 (2010)
19. Li, J., Lu, J.G., Chen, Y.: Robust decentralized control of perturbed fractional-order linear interconnected systems. *Comput. Math. Appl.* **66**(5), 844–859 (2013)
20. Moze, M., Sabatier, J., Oustaloup, A.: LMI tools for stability analysis of fractional systems. In: *20th ASME IEDTC/CIE*. Long Beach, California, USA (2005)
21. Sabatier, J., Moze, M., Farges, C.: On stability of fractional order systems. In: *Third IFAC Workshop on Fractional Differentiation and its Applications FDA'08*. Ankara, Turkey (2008)
22. Zhang, H., Feng, G., Yan, H., Chen, Q.: Observer based output feedback even-triggered control for consensus of multi-agent system. *IEEE Trans. Ind. Electron.* **61**(9), 4885–4894 (2014)
23. Zhang, Y., Gao, L., Tong, C.: On distributed reduced-order observer-based protocol for linear multi-agent consensus under switching topology. *Abstr. Appl. Anal.*, Article ID 793276 (2013)
24. Zhao, Y., Wen, G., Duan, Z., Xu, X., Chen, G.: A new observer type consensus protocol for linear multi-agent dynamical systems. *Asian J. Control* **15**(3), 1–12 (2013)
25. Horn, R.A., Johnson, C.R.: *Matrix Analysis*. Cambridge University Press, Cambridge (1985)
26. Brualdi, R.A., Ryser, H.J.: *Combinatorial Matrix Theory*. Cambridge University Press, Cambridge (1991)
27. Bernstein, D.: *Matrix Mathematics: Theory, Facts, and Formulas*. Princeton University Press, Princeton (2009)
28. Fuentes-Fernandez, R., Guijarro, M., Pajares, G.: A multi-agent system architecture for sensor networks. *Sensors* **9**(12), 10244–10269 (2009)

Chapter 10

Chaos Control and Anticontrol of Complex Systems via Parrondo's Game

Marius-F. Danca

Abstract In this chapter, we prove analytically and numerically aided by computer simulations, that the Parrondo game can be implemented numerically to control and anticontrol chaos of a large class of nonlinear continuous-time and discrete-time systems. The game states that alternating loosing gains of two games, one can actually obtain a winning game, i.e.: “losing + losing = winning” or, in other words: “two ugly parents can have beautiful children” (Zeilberger, on receiving the 1998 Leroy P. Steele Prize). For this purpose, the Parameter Switching (PS) algorithm is implemented. The PS algorithm switches the control parameter of the underlying system, within a set of values as the system evolves. The obtained attractor matches the attractor obtained by replacing the parameter with the average of switched values. The systems to which the PS algorithm based Parrondo's game applies are continuous-time of integer or fractional order ones such as: Lorenz system, Chen system, Chua system, Rössler system, to name just a few, and also discrete-time systems and fractals. Compared with some other works on switch systems, the PS algorithm utilized in this chapter is a convergent algorithm which allows to approximate any desired dynamic to arbitrary accuracy.

10.1 Introduction

In [34, 36], Parrondo et al. showed that alternating the loosing gains of two games, one can actually obtain a winning strategy with a positive gain, i.e.

$$\text{losing} + \text{losing} = \text{winning}. \quad (10.1)$$

M.-F. Danca (✉)

Department of Mathematics and Computer Science, Emanuel University of Oradea,
410597 Oradea, Romania
e-mail: danca@rist.ro

M.-F. Danca

Romanian Institute of Science and Technology, 400487 Cluj-Napoca, Romania

Since its discovery, this apparent contradiction has been known as Parrondo's paradox (or game, as we call in this work), becoming an active area of research for example in discrete-time ratchets [4], minimal Brownian ratchet [28], molecular transport [26], and so on. Parrondo's game is considered as game theory in the Blackwell sense [6] and in [2, 24] was extended from its original form to include player strategy. In [10, 11] a mechanism for pattern formation based on the alternation of two dynamics, is proposed. For a review of the history of Parrondo's paradox, developments, and connections to related phenomena, see [1].

This kind of alternation between weakness and strength, order and chaos, or losing and winning, can be found or produced in physical, biological, quantum, mathematical systems and in control theory, or even fractals, where combining processes may lead to counterintuitive dynamics. The apparently trivial phenomenon seems to be typical not only for theoretical systems but also in nature, where there are many interactions due to some accidental or intentional parameter switches. Even more, there is a belief that this kind of mechanisms could be used as a possible explanation of the origin of life [18].

If we replace in Parrondo's paradox the words "losing" with "chaos" and "winning" with "order" (as the opposite of chaos), then Parrondo's game can be written in the following form:

$$\text{chaos}_1 + \text{chaos}_2 = \text{order}, \quad (10.2)$$

where $\text{chaos}_{1,2}$ and order represent two chaotic dynamics and a regular dynamic respectively of a considered system. The form (10.2) of Parrondo's game is exploited in e.g. [3], where it is used to study the effects of combining different dynamics of two real systems, and also in [39, 40] where alternations between two dynamics of quadratic maps are investigated. In [15, 17], the study was extended to complex systems (fractals).

Relation (10.2) can be considered as a new kind of chaos control in the sense that by alternating two chaotic dynamics, it is possible to obtain a regular dynamic. Similarly, one can imagine an anticontrol-like scheme as

$$\text{order}_1 + \text{order}_2 = \text{chaos}. \quad (10.3)$$

A natural question is if it is possible to generalize Parrondo's game (10.2) in the sense that alternation between two dynamics in (10.2) is replaced with switches between $N > 2$ dynamics, i.e.

$$\text{chaos}_1 + \text{chaos}_2 + \cdots + \text{chaos}_N = \text{order}, \quad (10.4)$$

or

$$\text{order}_1 + \text{order}_2 + \cdots + \text{order}_N = \text{chaos}. \quad (10.5)$$

A positive answer is given in [39] for continuous time chaotic systems via the PS algorithm.

The goal of this chapter is to present a comprehensive account of the approaches used to define these chaos control-like and anticontrol-like algorithms, which are generalizations of Parrondo's paradox, via the PS algorithm.

10.2 Parameter Switching Algorithm

After presenting the general form of Parrondo's game, we describe the PS algorithm necessary to implement the Parrondo game. For this purpose, we have to choose a finite set of $N > 1$ parameters values, $\mathcal{P}_N = \{p_1, p_2, \dots, p_N\}$, inside which the algorithm switches the control parameter p as the considered continuous (discrete)-time system evolves. While for discrete-time systems, the algorithm simply switches p every m_i iterations, $i = 1, 2, \dots, N$, for the continuous-time systems, the time interval where the system is defined $I = [0, T]$, for $T > 0$, is partitioned in short time subintervals $I_{i,j}$, for $i = 1, 2, \dots, N$, $j = 1, 2, \dots$, each having length $m_i h$, h being a small real value (m_i being p_i “weights”), such that $I = \bigcup_j \bigcup_{i=1}^N I_{i,j}$ (see the sketch in Fig. 10.1 for $N = 2$). While the underlying Initial Value Problem (IVP) is numerically integrated, the algorithm switches successively p within \mathcal{P}_N in the subintervals $I_{i,j}$, $i = 1, 2, \dots, N$, $j = 1, 2, \dots$, i.e. in $I_{1,1}, I_{2,1}, \dots, I_{N,1}, I_{2,1}, I_{2,2}, \dots, I_{2,N}, I_{3,1}, \dots$ and so on, until the numerical integration ends.

For the sake of simplicity, hereafter the index j will be dropped unless necessary.

For continuous-time systems, the resulted “switched” attractor approximates the “averaged” attractor which is obtained if the parameter p is replaced with the average of the switched values, p^* (see Fig. 10.1):

$$p^* := \frac{\sum_{i=1}^N m_i p_i}{\sum_{i=1}^N m_i}, \quad p_i \in \mathcal{P}_N. \quad (10.6)$$

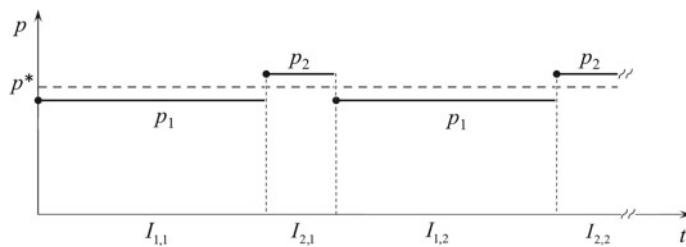


Fig. 10.1 Time subintervals $I_{i,j}$ and the piece-wise constant function p , for the case $N = 2$ (sketch)

10.2.1 PS Algorithm Applied to Continuous-Time Systems

Consider a class of systems modeled by the following IVP:

$$\dot{x}(t) = f(x(t)) + pAx(t), \quad t \in I = [0, T], \quad x(0) = x_0, \quad (10.7)$$

for $T > 0$, $x_0 \in \mathbf{R}^n$, $p \in \mathbf{R}$ the control parameter, $A \in L(\mathbf{R}^n)$ and $f : \mathbf{R}^n \rightarrow \mathbf{R}^n$ a nonlinear function.

The IVP (10.7) models a great majority of continuous nonlinear and autonomous dynamical systems depending on a single real control parameter p such as Lorenz system, Rösler system, Chen system, Lotka-Volterra system, Rabinovich-Fabrikant system, Hindmarsh-Rose system, Lü system, some classes of minimal networks, and many others. For example, for the Lorenz system

$$\begin{aligned}\dot{x}_1 &= a(x_2 - x_1), \\ \dot{x}_2 &= x_1(p - x_3) - x_2, \\ \dot{x}_3 &= x_1x_2 - cx_3,\end{aligned}\quad (10.8)$$

with $a = 10$, $c = 8/3$ and p the control parameter,¹

$$f(\mathbf{x}) = \begin{pmatrix} a(x_2 - x_1) \\ -x_1x_3 - x_2 \\ x_1x_2 - cx_3 \end{pmatrix}, \quad A = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

Let $p_h(t) = p(t)$ for any $h > 0$. Then, the “switching” equation (related to the PS algorithm) has the following form:

$$\dot{x}(t) = f(x(t)) + p_h(t)Ax(t), \quad t \in I = [0, T], \quad x(0) = x_0, \quad (10.9)$$

and the “average” equation, obtained for p replaced with p^* given by (10.6), is

$$\dot{\bar{x}}(t) = f(\bar{x}(t)) + p^*Ax(t), \quad t \in I = [0, T], \quad \bar{x}(0) = \bar{x}_0. \quad (10.10)$$

By applying the PS algorithm, the obtained switched solution of (10.9) will converge to the average solution of (10.10).

To approximate a desired solution, corresponding to some value p , we have to replace p^* with p in (10.6) and choose a set \mathcal{P}_N with the underlying weights m_i , $i = 1, 2, \dots, N$, such that (10.6) is verified. Next, by applying the PS algorithm with these ingredients, the obtained switched solution will approximate the searched (averaged) solution.

¹Also, a and c can be considered as control parameters to match to the form (10.7).

10.2.1.1 Convergence of the PS Algorithm

The following assumptions are made.

Assumption H1. f satisfies the usual Lipschitz condition:

$$|f(y_1) - f(y_2)| \leq L|y_1 - y_2|, \quad \forall y_{1,2} \in \mathbf{R}^n, \quad (10.11)$$

for some $L > 0$.

Assumption H2. The initial conditions x_0 and \bar{x}_0 belong to the same basin of attraction \mathcal{V} of the solution of (10.10).

Under the above assumptions, the convergence of the PS algorithm is given by the theorem

Theorem 10.1 ([21]) *Let $\|\cdot\|_0$ be the maximum norm on $C(I, \mathbf{R}^n)$. Under the above assumptions, it holds that*

$$|x(t) - \bar{x}(t)| \leq (|x_0 - \bar{x}_0| + h\|A\|\|\bar{x}\|_0 K) \times e^{(L+\|P\|_0\|A\|)T}, \quad (10.12)$$

for all $t \in [0, T]$, where

$$K := \max_{t \in [0, M_N]} \left| \int_0^t (P(s) - p^*) ds \right|.$$

Sketch of the proof:

From (10.9) and (10.10)

$$\begin{aligned} |x(t) - \bar{x}(t)| &\leq |x_0 - \bar{x}_0| + L \int_0^t |x(s) - \bar{x}(s)| ds + \left| \int_0^t (p_h(s) - p^*) ds \right| \|A\| \|\bar{x}\|_0 \\ &\quad + \|P\|_0 \|A\| \int_0^t |x(s) - \bar{x}(s)| ds = |x_0 - \bar{x}_0| + \|A\| \|\bar{x}\|_0 \left| \int_0^t (p_h(s) - p^*) ds \right| \\ &\quad + (L + \|P\|_0 \|A\|) \int_0^t |x(s) - \bar{x}(s)| ds \\ &\leq |x_0 - \bar{x}_0| + h\|A\|\|\bar{x}\|_0 K + (L + \|P\|_0 \|A\|) \int_0^t |x(s) - \bar{x}(s)| ds, \end{aligned}$$

and by Gronwall inequality [25], one obtains (10.12). \square

Next, adopt the following reasonable assumption regarding the notion of the (numerical) attractor utilized in this paper, necessary to implement numerically the PS algorithm.

Assumption H3. To every p value, for a given initial condition x_0 , there corresponds a unique solution and, therefore, a single numerical attractor, denoted by A_p , considered as a numerically approximation of its ω -limit set [22], after neglecting a sufficiently long transients.

The following theorem represents the main result concerning the PS algorithm for continuous-time systems.

Theorem 10.2 *Every attractor of the system (10.7) can be numerically approximated by the PS algorithm to arbitrary accuracy.*

Notation: Denote by A^* the “synthesized attractor”, obtained with the PS algorithm, and by A_{p^*} the “averaged attractor”, obtained for p replaced with p^* given by (10.6).

To obtain a desired attractor A_p corresponding to some value p , one has to replace in (10.6) p^* with p and choose an adequate set \mathcal{P}_N with underlying weights m_i , $i = 1, 2, \dots, N$, such that (10.6) is verified. Next, by applying the PS algorithm, the obtained (switched) attractor A^* will approximate the searched (averaged) attractor A_p .

Remark 10.1 The relation (10.6) is convex: if one denotes $\alpha_i = m_i / \sum_{k=1}^N m_k$, then $\sum_{i=1}^N \alpha_i = 1$, and $p^* = \sum_{i=1}^N \alpha_i p_i$. Therefore, the only necessary condition to approximate some attractor A_p is to choose \mathcal{P}_N such that $p \in (p_{\min}, p_{\max})$, with $p_{\min} = \min\{\mathcal{P}_N\}$ and $p_{\max} = \max\{\mathcal{P}_N\}$. Moreover, the convexity implies a robustness-like property of the PS algorithm: for every set \mathcal{P}_N , A^* will be situated “between” the attractors $A_{p_{\min}}$ and $A_{p_{\max}}$, with order being induced by the natural order of the real numbers in the parameter set \mathcal{P}_N .

Theorem 10.2 means that by choosing some value p , there always exists an attractor A_p (Remark 10.1) and a set of $N > 1$ parameters \mathcal{P}_N , such that $p^* = p \in (p_{\min}, p_{\max})$ with the underlying weights m_i , $i = 1, 2, \dots, N$, and p^* given by the relation (10.6).

Next, as stated by Theorem 10.2, A_{p^*} will be approximated by the attractor A^* , generated by the PS algorithm.

10.2.1.2 Numerical Implementation of the PS Algorithm

In order to implement numerically the PS algorithm, a numerical method for ODEs is necessary (for example, the standard Runge-Kutta method) with a fixed step size h . For the set \mathcal{P}_N with weights m_i , $i = 1, 2, \dots, N$, and a fixed step-size h , consider the PS algorithm in the following symbolic scheme:

$$[m_1 p_1, m_2 p_2, \dots, m_N p_N]. \quad (10.13)$$

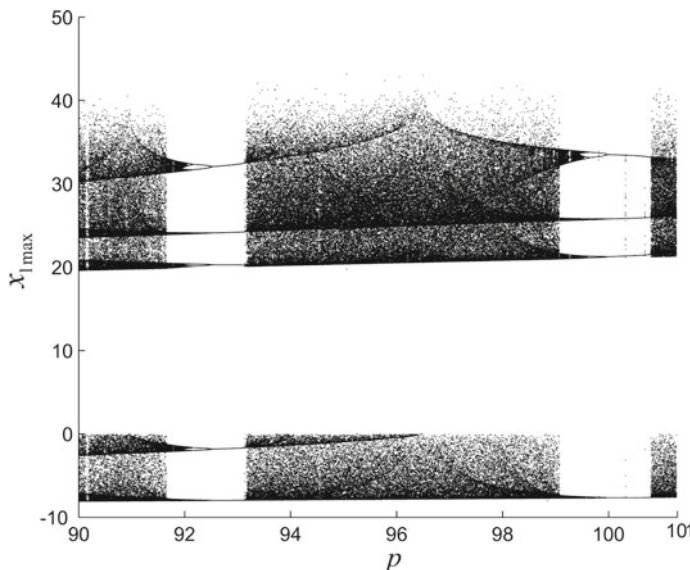
For example, if one wants to apply the PS algorithm on the set $\mathcal{P}_2 = \{p_1, p_2\}$ with weights $m_1 = 2$ and $m_2 = 1$, i.e. the scheme $[2p_1, 1p_2]$ applied with step size h , it means to do for the first two steps, $2h$, of integration of the underlying IVP, $p = p_1$, then for the next single step of size h , $p = p_2$, then for the next two

Table 10.1 The pseudocode of the PS algorithm

```

input :  $x_0, T, N, h, P_N, m_1, \dots, m_N$ 
output :  $x$ 
 $n := T/h, j := 1$ 
while  $j < n$ 
    for  $i = 1 : N$ 
        for  $k = 1 : m_i$ 
             $x_j \leftarrow \text{one step integration with } p = p_i$ 
             $j = j + 1$ 
        end
    end
end

```

**Fig. 10.2** Bifurcation diagram for the Lorenz system for $p \in [90, 101]$

steps, $p = p_1$, and so on, until the entire integration interval I is covered (see the pseudocode in Table 10.1).

Let consider the Lorenz system. The obtained switched and the averaged attractors are overplotted in the phase space and in time series. Also, whenever necessary, the Poincaré section is utilized. The integration time is $I = [0, 200]$ and $h = 0.002$. For the stable cycles, the transients were removed.

To have a general view of the parameter space wherefrom we have to peak the p values, a bifurcation diagram is shown in Fig. 10.2.

1. Next, we present the way in which the PS algorithm can be used to obtain (approximately) stable or chaotic attractors of *integer-order systems*.

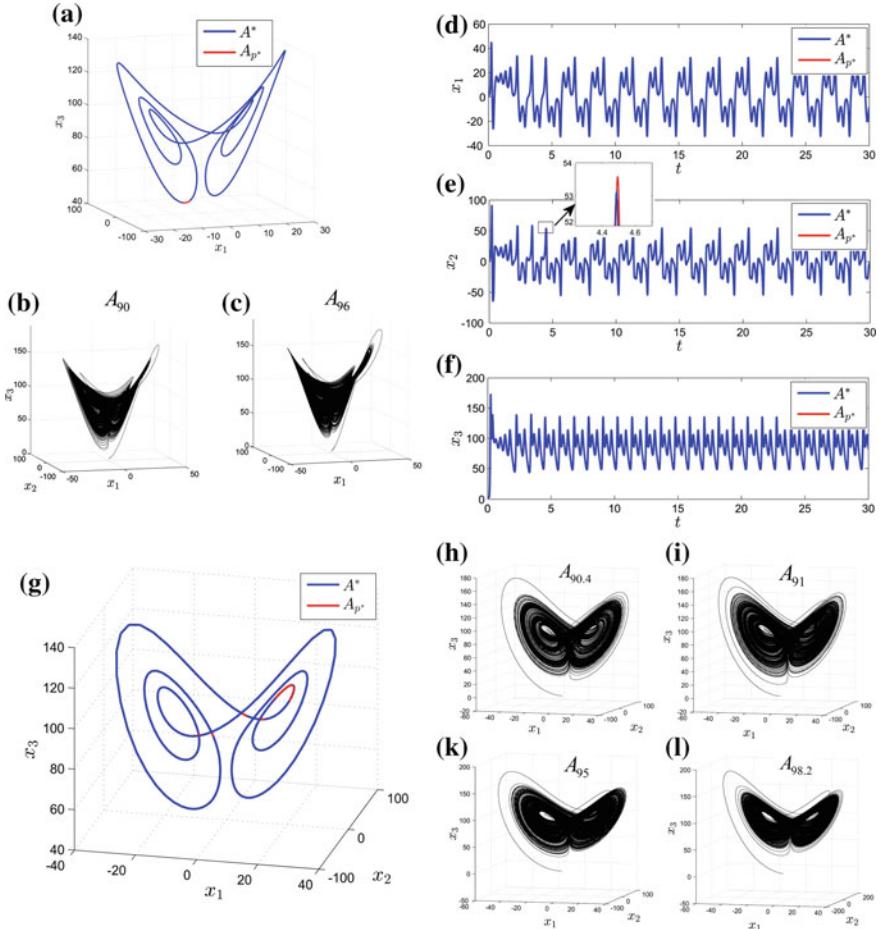


Fig. 10.3 Top Lorenz stable cycle corresponding to $p = 93$, obtained with the scheme $[1p_1, 1p_2]$ with $p_1 = 90$ and $p_2 = 96$ (Parrondo's chaos control game: $\text{chaos}_1 + \text{chaos}_2 = \text{order}$); **a** Phase overplot of the attractors A^* and A_{p^*} . **b, c** Underlying chaotic attractors A_{90} and A_{96} . **d–f** Overplot of attractors A^* and A_{p^*} time series. The enlarged view in Figure **e** reveals the inherently numerical errors; Bottom **g** Lorenz stable cycle corresponding to $p = 93$, obtained with the scheme $[2p_1, 1p_2, 1p_3, 1p_4]$, with $p_1 = 90.4$, $p_2 = 91$, $p_3 = 95$, $p_4 = 98.2$ (Parrondo's chaos control game: $\text{chaos}_1 + \text{chaos}_2 + \text{chaos}_3 + \text{chaos}_4 = \text{order}$)

- Suppose one wants to approximate the attractor corresponding to $p = 93$ (chosen in a periodic window, Fig. 10.2), which is a stable cycle. To do that, one can choose, for example $\mathcal{P}_2 = \{90, 96\}$, whose values belong to different chaotic windows in the parameter space (Fig. 10.3b, c) with weights $m_1 = m_2 = 1$, which when replaced in (10.6) gives the desired (average) value $p^* = (1 \times 90 + 1 \times 96)/2 = 93$. By applying the PS algorithm with the scheme $[1p_1, 1p_2]$, the obtained switched attractor A^* approximates the

averaged attractor A_{p^*} (Fig. 10.3a). A perfect match is also revealed by the overplotted time series in Fig. 10.3d–f. Even there exists an apparently perfect superposition, in the detail in Fig. 10.3e, one can see a relatively small difference between the two time series, due to the inherently numerical errors. Since in this example the attractors, corresponding to p_1 and p_2 , whose dynamics have been switched, are chaotic and the switched attractor is a regular motion, one can write in Parrondian words:

$$\text{chaos}_1 + \text{chaos}_2 = \text{order},$$

which represents Parrondo's game applied as a chaos control-like result.

- b. The same stable cycle can be obtained e.g. with the scheme $[2p_1, 1p_2, 1p_3, 1p_4]$, with $p_1 = 90.4, p_2 = 91, p_3 = 95, p_4 = 98.2$. Again, (10.6) gives $p^* = 93$ and the switched attractor A^* approximates the averaged attractor A_{p^*} (Fig. 10.3g). Since the attractors corresponding to $p_i, i = 1, 2, 3, 4$, are chaotic (Figs. 10.2 and 10.3h–i), the control-like Parrondo game is $\text{chaos}_1 + \text{chaos}_2 + \text{chaos}_3 + \text{chaos}_4 = \text{order}$.
 - c. The PS algorithm can be utilized for anticontrol too. For example, using the scheme $[1p_1, 1p_2]$ with $p_1 = 92$ and $p_2 = 100$ chosen in two periodic orbits (see Fig. 10.2 and Fig. 10.4a, b), one obtains the chaotic attractor A^* which approximates the stable attractor A_{p^*} with $p^* = 96$. Because one should use an infinity time to generate the chaotic attractors, the inherently finite-time approximation is less accurate than that for chaos control, as can be seen in Fig. 10.4 c. However, the shapes of A^* and A_{p^*} look similar, as indicated also by the Poincaré section with the plane $x_3 = 100$ (Fig. 10.4d). In Parrondian words, the anticontrol result can be written $\text{order}_1 + \text{order}_2 = \text{chaos}$.
2. The PS algorithm applies also to *fractional-order systems*²

Consider the Chen system of fractional-order [13, 29] in the following form:

$$\begin{aligned} D_*^{0.92}x_1 &= p(x_2 - x_1), \\ D_*^{0.95}x_2 &= (3.65 - p)x_1 + 3.65x_2 - x_1x_3, \\ D_*^{0.90}x_3 &= x_1x_2 - 0.3x_3, \end{aligned} \quad (10.14)$$

where D_*^q denotes the the Caputo differential operator of order q (see e.g. [12, 33, 37]). The numerical method used here to integrate the system is the Grünwald-Letnikov method for fractional differential equations (see e.g. [5, 30, 41]). For $p_1 = 4.243$ and $p_2 = 4.302$, the system behaves chaotically (Fig. 10.5a, b) and with the scheme $[3p_1, 1p_2]$ one obtains $p^* = 4.25775$ for which the system is stable. By applying the PS algorithm, the switched attractor A^* matches perfectly

²There exists no convergence result so far. However, intensively numerical tests reveal, like in the considered example, a good match between the switched attractor and the averaged attractor in the case of fractional-order systems.

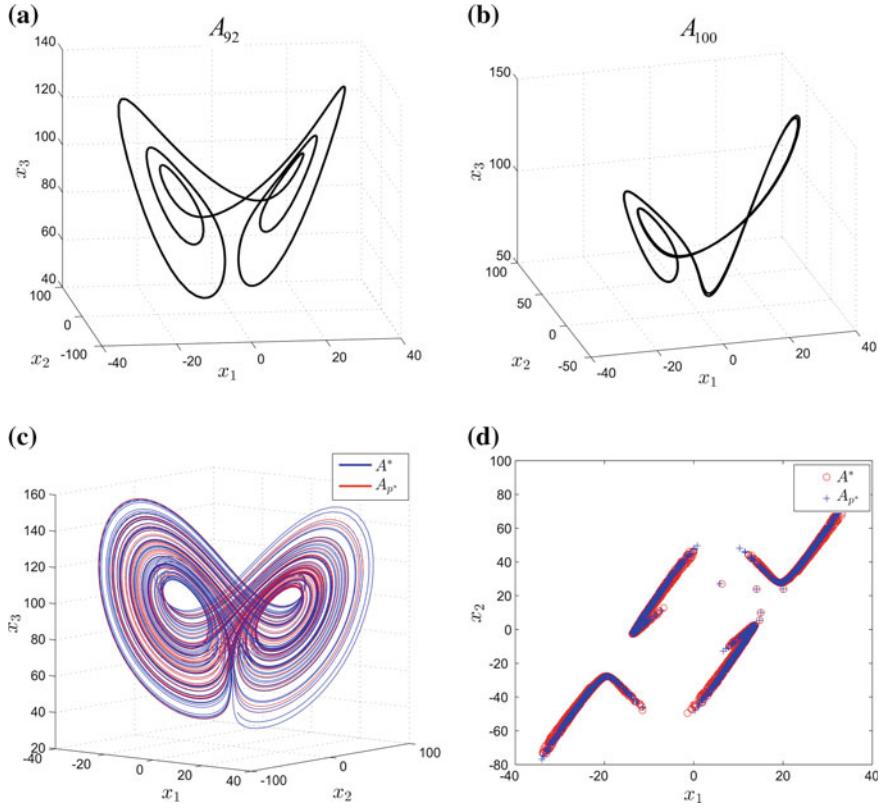


Fig. 10.4 Lorenz chaotic attractor corresponding to $p = 96$, obtained with the scheme $[1p_1, 1p_2]$ with $p_1 = 92$ and $p_2 = 100$ (Parrondo's anticontrol game: $order_1 + order_2 = chaos$). **a, b** Underlying stable cycles A_{92} and A_{100} ; **c** Phase overplot of the attractors A^* and A_{p^*} ; **d** Poincaré section with $x_3 = 100$ through the overplot attractors A^* and A_{p^*}

the averaged attractor A_{p^*} (Fig. 10.5c), and this chaos control-like Parrondo game reads $chaos_1 + chaos_2 = order$.

In the above examples, the scheme (10.13) is implemented periodically: the values of p take successively the values p_1 for m_1 times, then p_2 for m_2 times, and so on until p_N for m_N times, after which it repeats. However, the order of p_i with the underlying weight m_i can be taken *randomly* by using, for example, some random uniformly distributed sequence³ of values p_i . The averaged value, denoted \bar{p}^* , has to be determined now by the following relation:

$$\bar{p}^* := \frac{\sum_{i=1}^N m'_i p_i}{\sum_{i=1}^N m'_i}, \quad p_i \in \mathcal{P}_N, \quad (10.15)$$

³E.g. the *pseudorandom* function, found in all dedicated software.

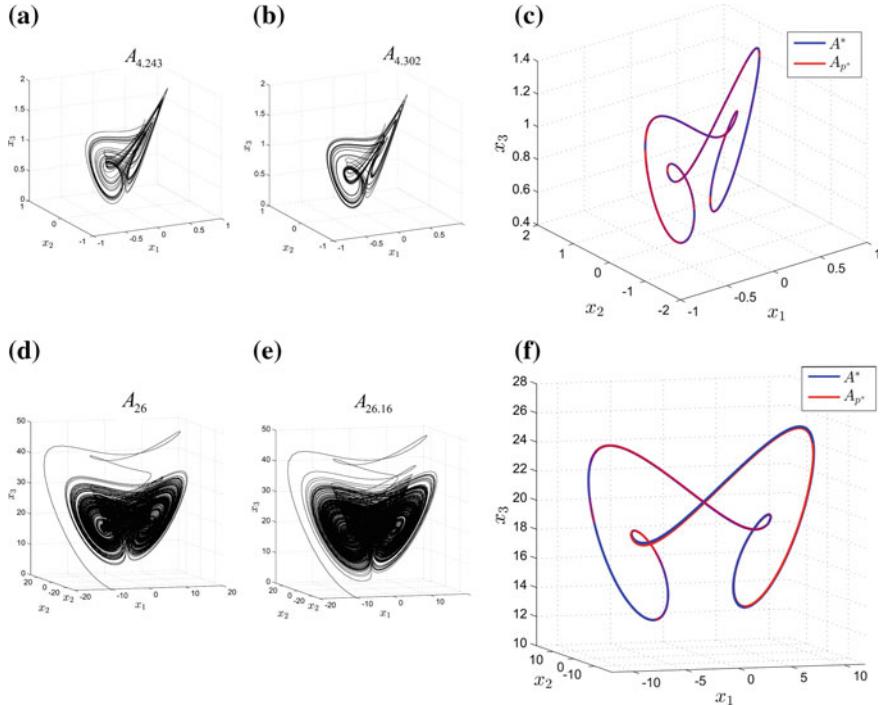


Fig. 10.5 **Top** Stable cycle of the fractional-order Chen system (10.14), corresponding to $p = 4.25775$ (Parrondo's chaos control game: $\text{chaos}_1 + \text{chaos}_2 = \text{order}$); **a, b** Underlying chaotic attractors $A_{4,243}$ and $A_{4,302}$; **c** Phase overplot of the attractors A^* and A_{p^*} ; **Bottom** Stable cycle of the Chen system of integer order (10.16) corresponding to $p = 26.08$ obtained with randomly applied scheme $[1p_1, 1p_2]$; **d, e** Underlying chaotic attractors A_{26} and $A_{26,16}$; **f** Phase overplot of the attractors A^* and A_{p^*}

where, m'_i are the total number of switchings of p_i when the integration ends. After a sufficiently large integration interval I , $\bar{p}^* \approx p^*$. However, in this case, for the considered example, supplementary precautions should be considered, such as the dispersal of p values in the parameter space, which have to be close to p^* . Also, the integration interval has to be larger and the step-size h smaller.

Consider the Chen system of integer order in the following form

$$\begin{aligned}\dot{x}_1 &= 35(x_2 - x_1), \\ \dot{x}_2 &= (p - 35)x_1 - x_2x_3 + px_2, \\ \dot{x}_3 &= x_1x_2 - 3x_3,\end{aligned}\tag{10.16}$$

and suppose one wants to obtain the stable cycle corresponding to $p = 26.08$ [16] by the scheme $[1p_1, 1p_2]$ with $p_1 = 26$ and 26.16 , generating chaotic attractors (Fig. 10.5d, e). With the step size $h = 0.0005$ and the integration interval

$I = [0, 800]$, the PS algorithm approximates the stable cycle (Fig. 10.5f). Now, the relatively small differences between the two attractors are clearer.

10.2.2 PS Algorithm Applied to Maps

As proved analytically in Sect. 10.2.1.1 by applying Parrondo's game to continuous-time systems, the switched solution obtained with the PS algorithm converges to the averaged solution. However, for the discrete systems, things are different.

Consider the following discrete variant of (10.7):

$$x_{k+1} = f(x_k) + q_k Ax_k,$$

where $x_0 \in \mathbf{R}^n$, $f : \mathbf{R}^n \rightarrow \mathbf{R}^n$ satisfies the Lipschitz condition, and $\{q_k\}_{k \in \mathbb{N}}$, $q_k = p_i$ for $k \in [M_{i-1} + 1, M_i]$, $M_0 = 0$, $M_i := \sum_{j=1}^i m_j$, $p_i \in \mathcal{P}_N$, $1 \leq i \leq N$, and $T := M_N$, is a T -periodic piecewise constant sequence. Then, there is no any relationship with the average equation

$$x_{k+1} = f(x_k) + p^* Ax_k,$$

where p^* is given by (10.6), such as for the case of continuous-time systems.

However, for the following discrete version of the PS algorithm:

$$x_{k+1} = x_k + h(f(x_k) + q_k Ax_k),$$

there exists an averaged form:

$$\bar{x}_{k+1} = \bar{x}_k + h(f(\bar{x}_k) + p^* A\bar{x}_k),$$

where the averaging theory applies [14].

Even Theorem 10.1 does not apply to the most-known discrete-time systems (like the logistic map), or to complex systems (like fractals), but the PS algorithm still works as chaos-control like and anticontrol-like tasks, for which quite intriguing results as can be seen as in the next section.

10.2.2.1 PS Algorithm Applied to the Logistic Map

Apply the PS algorithm to the logistic map $f : [0, 1] \rightarrow [0, 1]$, $f(x) = px(1 - x)$, $p \in [0, 4]$, in the following simplest form⁴:

$$x_{k+1} = q_k x_k (1 - x_k), \quad k = 0, 1, \dots \quad (10.17)$$

⁴In [3, 23], some particular forms of switches are used to study the behavior of alternated orbits for the more accessible quadratic (Mandelbrot) map $x_{k+1} = x_k^2 + p$.

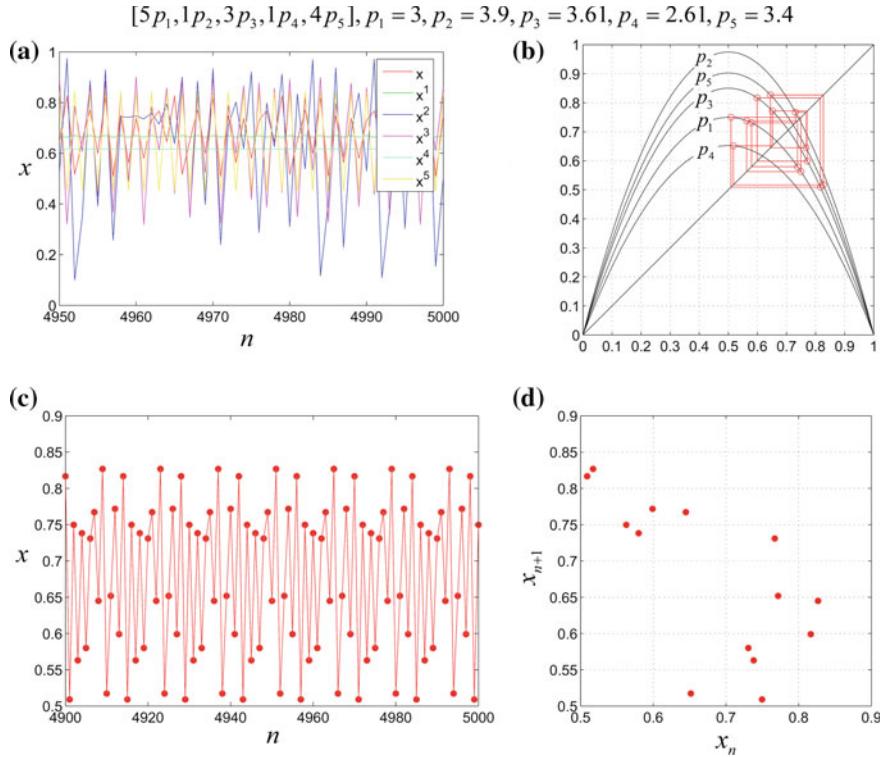


Fig. 10.6 Stable cycle obtained using the scheme $[5p_1, 1p_2, 3p_3, 1p_4, 4p_5]$ with $\mathcal{P}_5 = \{3, 3.9, 3.61, 2.61, 3.4\}$ applied to the logistic map (Parrondo's chaos-control game: $chaos_1 + chaos_2 + chaos_3 + chaos_4 + chaos_5 + order_1 = order$); **a** Orbits of the underlying dynamics corresponding to p_i , $i = 1, 2, \dots, 5$; **b** Cobweb indicating the multiple periods of the stable cycle; **c** Time series of the controlled orbit; **d** First return map

To analyze the numerical results, one can use time series, cobweb and first-return map, have been utilized with q_k defined as above: $q_k = p_i$ for $k \in [M_{i-1} + 1, M_i]$, $1 \leq i \leq N$.

This time, with the PS algorithm one can obtain stable orbits which are different from those of the logistic map [14]. Therefore, the PS algorithm can be used to control chaos or obtain chaoticization. By choosing empirically the weights m_i and \mathcal{P}_N , it is possible to control the chaotic behavior of the logistic map. As verified numerically in [14], there exists a positive probability to realize chaos control by using (the generalized) Parrondo's game.

1. For example, choosing the scheme $[5p_1, 1p_2, 3p_3, 1p_4, 4p_5]$ with $\mathcal{P}_5 = \{3, 3.9, 3.61, 2.61, 3.4\}$, one obtains the following Parrondo's game for chaos control (Fig. 10.6): $order_1 + chaos_2 + chaos_3 + order_2 + order_3 = order$. The dynamics corresponding to p_i , $i = 1, 2, 3, 4, 5$, are plotted in Fig. 10.6 a. In this case, $order$ represents a stable orbit, different from but similar to any of the possible orbits

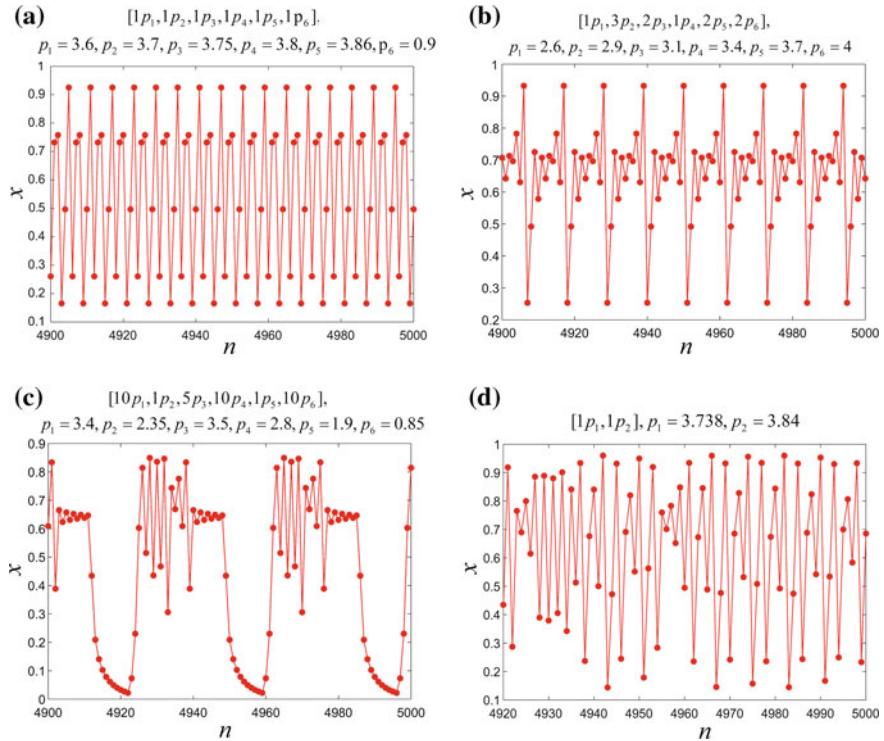


Fig. 10.7 Other chaos control and anticontrol of the logistic map; **a** Chaos control with $[1p_1, 1p_2, 1p_3, 1p_4, 1p_5, 1p_5, 1p_6]$ with $p_1 = 3.6, p_2 = 3.7, p_3 = 3.75, p_4 = 3.8, p_5 = 3.86$ and $p_6 = 0.9$ (Parrondo's chaos control game: $order_1 + order_2 + order_3 + order_4 + chaos_1 + chaos_2 = order$); **b** Chaos control with $[1p_1, 3p_2, 2p_3, 1p_4, 2p_5, 2p_6]$ with $p_1 = 2.6, p_2 = 2.9, p_3 = 3.1, p_4 = 3.4, p_5 = 3.7, p_6 = 4$ (Parrondo's chaos control game: $order_1 + order_2 + order_3 + order_4 + chaos_1 + chaos_2 = order$); **c** Chaos control with $[10p_1, 1p_2, 5p_3, 10p_4, 1p_5, 10p_6]$, $p_1 = 3.4, p_2 = 2.35, p_3 = 3.5, p_4 = 2.8, p_5 = 1.9, p_6 = 0.85$ (Parrondo's chaos control game: $order_1 + order_2 + order_3 + order_4 + order_5 + order_6 = order$); **d** Anticontrol with $[1p_1, 1p_2]$ with $p_1 = 3.738$ and $p_2 = 3.84$ (Parrondo's anticontrol game: $chaos_1 + order_1 = chaos_2$)

of the logistic map, revealed by the cobweb, time series and first return map (Fig. 10.6 b, c and d, respectively).

2. By using the scheme $[1p_1, 1p_2, 1p_3, 1p_4, 1p_5, 1p_5, 1p_6]$ with $p_1 = 3.6, p_2 = 3.7, p_3 = 3.75, p_4 = 3.8, p_5 = 3.86$ and $p_6 = 0.9$, one obtains the stable orbit plotted in Fig. 10.7 a. In this case, chaos control is implemented by Parrondo's game: $chaos_1 + chaos_2 + chaos_3 + chaos_4 + chaos_5 + order_1 = order$.
3. The stable orbit plotted in Fig. 10.7 b is obtained by the scheme $[1p_1, 3p_2, 2p_3, 1p_4, 2p_5, 2p_6]$ with $p_1 = 2.6, p_2 = 2.9, p_3 = 3.1, p_4 = 3.4, p_5 = 3.7, p_6 = 4$. In this case, the Parrondo game has the following form: $order_1 + order_2 + order_3 + order_4 + chaos_1 + chaos_2 = order$.

4. The periodic bursts in Fig. 10.7 c [14] are obtained by the scheme $[10p_1, 1p_2, 5p_3, 10p_4, 1p_5, 10p_6]$, with $p_1 = 3.4$, $p_2 = 2.35$, $p_3 = 3.5$, $p_4 = 2.8$, $p_5 = 1.9$, $p_6 = 0.85$ and Parrondo's game is: $order_1 + order_2 + order_3 + order_4 + order_5 + order_6 = order$.
5. If one uses the scheme $[1p_1, 1p_2]$ with $p_1 = 3.738$ and $p_2 = 3.84$, the PS algorithm simulates the Parrondo game to model the anticontrol of chaos: $chaos_1 + order_1 = chaos_2$ (Fig. 10.7 d).

10.2.2.2 PS Algorithm Applied to Fractals

In [17] the PS algorithm is used to alternate two different dynamics of the quadratic complex map $z_{n+1} = z_n^2 + c_i$ to prove that the obtained sets, called alternated Julia sets, can be connected, disconnected, or totally disconnected verifying the Fatou-Julia theorem [20, 27] in the case of polynomials of degree greater than two.

Because in this case one deals with a set of two values, c_1 and c_2 , one operates with “alternations”, not switchings.

As is known, for a complex polynomial $P : \mathbf{C} \rightarrow \mathbf{C}$ of degree $d \geq 2$, infinity is a superattracting fixed point. If one denotes by $\mathcal{A}(\infty)$ the attraction basin of ∞ for the polynomial P , $A(\infty) = \{z \in C | P^{\circ n} \rightarrow \infty\}$, then the *filled Julia set* of P is the set $K = C \setminus \mathcal{A}(\infty)$. The boundary of the filled Julia set is called the *Julia set*, where chaotic dynamics occur.

The connectivity properties of the Julia set are in a relationship with the dynamical properties about its finite critical points (Fatou-Julia Theorem [20, 27]): *The Julia set is connected if and only if all the critical orbits are bounded; and the set is totally disconnected, a Cantor set, if (but not only if) all the critical orbits are unbounded.* In [9, 38], the theorem was completed as follows: *For a polynomial with at least one critical orbit unbounded, the Julia set is totally disconnected if and only if all the bounded critical orbits are aperiodic.*

The alternated Julia sets K_{c_1, c_2} are the set of points in the complex plane with bounded orbits when one iterates the alternated system

$$P_{c_1 c_2} : z_{n+1} = \begin{cases} z_n^2 + c_1, & n \text{ even}, \\ z_n^2 + c_2, & n \text{ odd}. \end{cases}$$

The generated orbit is

$$\begin{aligned} z_0, \\ z_1 &= z_0^2 + c_1, \\ z_2 &= (z_0^2 + c_1)^2 + c_2, \\ z_3 &= ((z_0^2 + c_1)^2 + c_2)^2 + c_1, \\ &\dots \end{aligned}$$

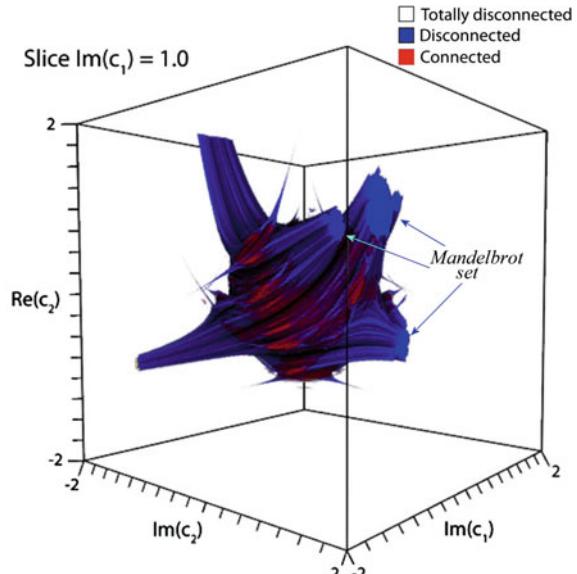
In a similar way one can define the alternated filled Julia set K_{c_2, c_1} , which has the same shape with the alternated filled Julia set K_{c_1, c_2} , as being the set of points in the complex plane with bounded orbits when one iterates the alternated system

$$P_{c_2 c_1} : z_{n+1} = \begin{cases} z_n^2 + c_2, & n \text{ even}, \\ z_n^2 + c_1, & n \text{ odd}. \end{cases}$$

In [17], it is proved that the alternated Julia sets verify the Fatou-Julia theorem in the case of complex polynomials of degree greater than two.

As known, the Julia set is totally disconnected if c does not belong to the Mandelbrot set [7, 8, 32]. However, in [17] it is proved that the alternated Julia sets can be connected, disconnected or totally disconnected. Because the totally disconnected sets, disconnected sets and connected form a four-dimensional body (it depends on four real variables: $\operatorname{Re}(c_{1,2})$ and $\operatorname{Im}(c_{1,2})$), to study computationally the connectivity problem, one has to fix some of these variables, and scroll the others within some domain. In other words, to obtain a three-dimensional views (of the four existing objects), one has to slice the four-dimensional body with one of the four planes $\operatorname{Re}(c_{1,2}) = ct$, $\operatorname{Im}(c_{1,2}) = ct$ (see the volume rendering [19, 31] in Fig. 10.8, where a three-dimensional view is obtained by sectioning the body with the plane $\operatorname{Im}(c_1) = 1$). To obtain two dimensional views, two planes sections (slices) are necessary.

Fig. 10.8 Three-dimensional view of the connectivity body of the alternated Julia sets, obtained with the section with $\operatorname{Im}(c_1) = 1$. The white region (body's exterior) indicates the points for which the alternated Julia sets are totally disconnected, the blue regions indicate the disconnectedness while the red regions the connectedness



For example, if one considers the planar section with $c_2 = -0.1562 + 1.0320i$ and $c_1 \in [-0.176, -0.136] \times [1.012, 1052]$ (Fig. 10.9 a), the filled Julia set correspond-

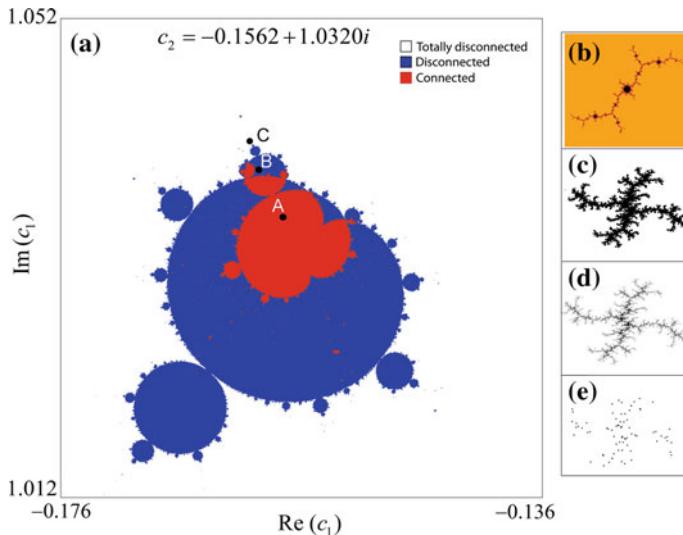


Fig. 10.9 **a** Section through the three-dimensional body, obtained by alternating Julia sets with $c_2 = -0.1562 + 1.0320i$ and $c_1 \in [-0.176, -0.136] \times [1.012, 1.052]$; **b** Totally disconnected filled Julia Julia set corresponding to $c = -0.1562 + 1.0320i$; **c** Connected alternated filled Julia set corresponding to the point *A*; **d** Disconnected alternated Julia set corresponding to the point *B*; **e** Totally disconnected alternated Julia set corresponding to then point *C*

ing to $c_2 = -0.1562 + 1.0320i$ is a totally disconnected set (Fig. 10.9 b), while the alternated Julia sets for $c_2 = -0.1562 + 1.0320i$ and c_1 considered in the connected region (point *A*) is a connected set (Fig. 10.9 c), for $c_2 = -0.1562 + 1.0320i$ and c_1 considered in the disconnected region (point *B*) is a disconnected set (Fig. 10.9 d), and for $c_2 = -0.1562 + 1.0320i$ and c_1 considered in the white region (point *C*) is a totally connected set (Fig. 10.9 e).

Remark 10.2 Representing graphically the three-dimensional connectivity bodies, a remarkable property was revealed in [17]: as known, the Mandelbrot set is the set of all c values for which each (classical) Julia set is connected. However, the “ends” of the three-dimensional body shown in Fig. 10.8, indicate a new and intriguing property: it is the set of all parameter values, for which each corresponding alternated Julia set is *disconnected* form Mandelbrot sets (the blue points in Fig. 10.9 a). By using special algorithms to draw fractals, one can prove that the apparently separated parts (dots) of connectivity and disconnectivity are in reality connected to their body [35, Chap. 4].

10.3 Conclusion

In this chapter, we have presented the approach of a generalization of Parrondo's game, implemented for both continuous-time and discrete-time systems, via the PS algorithm. Thus, by applying the PS algorithm, the forms of Parrondo's para-

dox game read $chaos_1 + chaos_2 + \dots + chaos_N = order$, for $N \geq 2$, or $order_1+order_2+\dots+order_N = chaos$, acting like chaos-control like or anticontrol-like behaviors. Also, combinations of ordered and chaotic motions can lead to chaos-control like and anticontrol-like results. These generalizations of Parrondo's game, applied as chaos control or anticontrol schemes have been used here to Lorenz system, Chen systems of integer and fractional order, the logistic map, and also fractals (alternated Julia sets). While for the continuous-time systems, the convergence of the PS algorithm has been proved analytically, but for the fractional-order systems, the convergence has been verified only numerically. Also, for the logistic map, the PS algorithm generates different orbits from the existing orbits, Parrondo's paradox has been implemented to realize chaos control and anticontrol. The apparently paradoxical result obtained with the PS algorithm applied to continuous systems, resides in the linearly dependence on the parameter p in the underlying IVP. Although this particularity seems to be restrictive, it characterizes most-known continuous systems. One of the most interesting new property, revealed by the PS algorithm, is the fact that the Mandelbrot set seems to be not only the set of all complex points for which the Julia sets are connected, but also the set of all complex points for which the alternated Julia sets are disconnected. With the PS algorithm, every attractor of a considered system can be generated (approximated), but due to some objective reasons, one cannot set some parameter values. The PS algorithm can be used as a possible explanation of the strange dynamics of some systems where switchings between the underlying dynamics occur, either periodically or randomly.

References

1. Abbot, D.: Assymmetry and disorder: a decade of parrondo's paradox. *Fluct. Noise Lett.* **9**, 129–156 (2010)
2. Abbott, D., Davies, P.C.W., Shalizi, C.R.: Order from disorder: the role of noise in creative processes: a special issue on game theory and evolutionary processes -overview. *Fluct. Noise Lett.* **7**(2), 1–12 (2003)
3. Almeida, J., Peralta-Salas, D., Romera, M.: Can two chaotic systems give rise to order. *Phys. D* **200**, 124–132 (2005)
4. Amengual, P., Allison, A., Toral, R., Abbott, D.: Discrete-time ratchets, the Fokker-Planck equation and parondo's paradox. *Proc. R. Soc. Lond. Ser. A Math. Phys. Eng. Sci.* **460**, 2269–2284 (2004)
5. Baleanu, D., Diethelm, K., Scalas, E., Trujillo, J.J.: Fractional Calculus Models and Numerical Methods. Series on Complexity, Nonlinearity and Chaos. World Scientific, Singapore (2012)
6. Blackwell, D., Girshick, M.A.: Theory of Games and Statistical Decisions. John Wiley, New York (1954)
7. Blanchard, P., Devaney, R., Keen, L.: Complex dynamics and symbolic dynamics, symbolic dynamics and Its Applications. In: Williams, S.G. (ed.) Proceedings of Symposium in Applied Mathematics 60, pp. 37–60 (2004)
8. Blanchard, P.: Disconnected Julia sets. In: Barnsley, M., Demko, S. (eds.) Chaotic Dynamics and Fractals, pp. 181–201. Academic Press, San Diego (1986)
9. Branner, B., Hubbard, J.H.: Iteration of cubic polynomials, part II: patterns and parapatterns. *Acta Math.* **169**, 229–325 (1992)

10. Buceta, J., Lindenberg, K., Parrondo, J.M.R.: Spatial patterns induced by random switching. *Fluct. Noise Lett.* **2**(1), L21–L29 (2002)
11. Buceta, J., Lindenberg, K., Parrondo, J.M.R.: Stationary and oscillatory spatial patterns induced by global periodic switching. *Phys. Rev. Lett.* **88**, 024103 (2002)
12. Caputo, M.: Linear models of dissipation whose Q is almost frequency independent—II. *Geophys. J. R. Astron. Soc.* **13**, 529–539 (1967) (reprinted in *Fract. Calc. Appl. Anal.* **10**(3), 309–324 (2007))
13. Li, C., Chen, G.: Chaos in the fractional order Chen system and its control. *Chaos Soliton. Fract.* **22**(3), 549–554 (2004)
14. Danca, M.F., Fečkan, M., Romera, M.: Generalized Form of Parrondo's Paradoxical Game with Applications to Chaos Control. *Int. J. Bifurcat. Chaos.* 2014, Accepted
15. Danca, M.F., Bourke, P., Romera, M.: Graphical exploration of the connectivity sets of alternated Julia sets; M, the set of disconnected alternated Julia sets. *Nonlinear Dynam.* **73**, 1155–1163 (2013)
16. Danca, M.F., Wallace, K.S.: Tang, Chen, G.: A switching scheme for synthesizing attractors of dissipative chaotic systems. *Appl. Math. Comput.* **201**(1–2), 650–667 (2008)
17. Danca, M.F., Romera, M., Pastor, G.: Alternated Julia sets and connectivity properties. *Int. J. Bifurcat. Chaos.* **19**(6), 2123–2129 (2009)
18. Davies, P.C.W.: Physics and life: The Abdus Salam Memorial Lecture. 6th Trieste Conference on Chemical Evolution. Eds: J. Chela-Flores, T. Tobias, and F. Raulin. Kluwer Academic Publishers, 13–20 (2001)
19. Drebin, R.A., Carpenter, L., Hanrahan, P.: Volume rendering. In: Proceedings of the 15th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH 88), Vol. 22 pp. 65–74 (1988)
20. Fatou, P.: Sur les équations fonctionnelles. *B. Soc. Math. Fr.* **47** (1919) 161–271; **48** (1920) 33–94; **48** (1920) 208–314. <http://smf.emath.fr/Publications/Bulletin/>
21. Fečan, M., Danca, M.F.: Note on a Parameter Switching Method for Nonlinear ODEs. *Math. Slovaca.* Accepted (2014)
22. Foias, C., Jolly, M.S.: On the numerical algebraic approximation of global attractors. *Nonlinearity* **8**, 295–319 (1995)
23. Fulai, W.: Improvement and empirical research on chaos control by theory of “chaos+chaos=order”. *CHAOS* **22**, 043145 (2012)
24. Groeber, P.: On Parrondos games as generalized by behrends. *Lect. Notes Cont. Inform. Sci.* **341**, 223–230 (2006)
25. Hale, J.K.: Ordinary Differential Equations. Dover Publications, New York (2009) (first published: John Wiley & Sons (1969))
26. Heath, D., Kinderlehrer, D., Kowalczyk, M.: Discrete and continuous ratchets: from coin toss to molecular motor. *Discrete Cont. Dyn.-B.* **2**, 153–167 (2002)
27. Julia, G.: Mémoire sur l'itération des fonctions rationnelles. *J. Math. Pure Appl.* **8**, 47–245 (1918)
28. Lee, Y., Allison, A., Abbott, D., Stanley, H.E.: Minimal Brownian ratchet: an exactly solvable model. *Phys. Rev. Lett.* **91**, 220601 (2003)
29. Li, C., Peng, G.: Chaos in Chen's system with a fractional order. *Chaos Soliton. Fract.* **22**(2), 443–450 (2004)
30. Li, C., Zeng, F.: Finite difference methods for fractional differential equations. *Int. J. Bifurcat. Chaos.* **22**(4), 1230014 (2012)
31. Limaye, A.: Drishti: a volume exploration and presentation tool. In: Proceedings SPIE 8506, Developments in X-Ray Tomography VIII, 85060X (2012)
32. Mandelbrot, B.B.: The Fractal Geometry of Nature. Freeman, New York (1977)
33. Oldham, K.B., Spanier, J.: The Fractional Calculus: Theory and Applications of Differentiation and Integration of Arbitrary Order. Academic Press, New York (1974)
34. Parrondo, J.M.R., Harmer, G.P., Abbott, D.: New paradoxical games based on Brownian ratchets. *Phys. Rev. Lett.* **85**, 5226–5229 (2000)
35. Peitgen, H.O., Saupe, D.: The Science of Fractal Images. Springer, New York (1988)

36. Percus, O.E., Percus, J.K.: Can two wrongs make a right? Coin-tossing games and parrondo's paradox. *Math. Intell.* **24**, 68–72 (2002)
37. Podlubny, I.: Fractional Differential Equations. Academic Press, San Diego (1999)
38. Qiu, W., Yin, Y.: Proof of the Branner-Hubbard conjecture on Cantor Julia sets. *Sci. China Ser. A.* **52**(1), 45–65 (2009)
39. Romera, M., Pastor, G., Danca, M.F., Martin, A., Orue, A.B., Montoya, F.: Alternate iteration of a quadratic map. *Int. J. Bifurcat. Chaos*. Accepted (2014)
40. Romera, M., Small, M., Danca, M.F.: Deterministic and random synthesis of discrete chaos. *Appl. Math. Comput.* **192**(1), 283–297 (2007)
41. Scherer, R., Kalla, S.L., Tang, Y., Huang, J.: The Grünwald-Letnikov method for fractional differential equations. *Comput. Math. Appl.* **62**(3), 902–917 (2011)

Chapter 11

Collective Behavior Coordination with Predictive Mechanisms

Hai-Tao Zhang, Zhaomeng Cheng, Ming-Can Fan and Yue Wu

Abstract In natural flocks/swarms, it is very appealing that low-level individual intelligence and communication can yield advanced coordinated collective behaviors such as congregation, synchronization and migration. Firstly, we seek to understand the role of predictive mechanisms in the forming and evolving of flocks/swarms by using both numerical simulations and mathematical analyses. Secondly, by incorporating some predictive mechanism into a few pinning nodes, we show that convergence procedure to consensus can be substantially accelerated in networks of interconnected dynamic agents while physically maintaining the network topology. Such an acceleration stems from the compression mechanism of the eigenspectrum of the state matrix conferred by the predictive mechanism. Thirdly, some model predictive control protocols are developed to achieve consensus for a class of discrete-time double-integrator multi-agent systems with input constraints. Associated sufficient conditions such as that the proximity net has a directed spanning tree and that the sampling period is sufficiently small are proposed. Moreover, the control horizon is extended to larger than one, which endows sufficient degrees of freedom to accelerate the convergence to consensus.

H.-T. Zhang (✉) · Z. Cheng · Y. Wu

The Key Lab of Image Processing and Intelligent Control School of Automation and
The State Key Lab of Digital Manufacturing Equipment and Technology,
Huazhong University of Science and Technology,
Wuhan 430074, People's Republic of China
e-mail: zht@mail.hust.edu.cn

Z. Cheng
e-mail: zmcheng@hust.edu.cn

Y. Wu
e-mail: wuyue_a@126.com

M.-C. Fan
Department of Mathematics, Huizhou University, Huizhou 516007,
Guangdong, People's Republic of China
e-mail: mingcan.fan@gmail.com

11.1 Injection Predictive Protocol to Flocking Models

The objective of this section is to reveal the important role of the predictive mechanisms in the emergence of collective behaviors and to design autonomous and reliable predictive protocols for industrial multi-agent systems. We will present some recently developed theoretical and numerical tools for modeling, analyzing and designing predictive motion protocols for flocks/swarms, both with and without leaders. We will pay special attention to the development of design methodologies that provide industrial multi-agent systems with provably correct cooperative predictive strategies, the characterization of the improvement of the system's collective behavior induced by the predictive capability, and the effective economization of the communication cost owing to such capability.

We will approach the problem in two stages. Firstly, we emphasize the role of predictive mechanisms for some specific types of flocks/swarms with leaders, namely the A/R [6] and Vicsek [17] models. More specifically, a predictive protocol is designed based on the Wattz-Strogatz (WS) small-world connection model [19]. This preliminary study shows that the intelligent predictive capability not only improves the cohesive and formative collective behavior but also reduces the long-range communication cost inside the flock. The power of predictive mechanisms is thus initially demonstrated.

The second focus is the decentralized predictive mechanisms in networks without leaders. To develop a decentralized predictive protocol, the question is essential whether the information provided by the local neighbors of an individual is enough to estimate their future states. The proof of this result is established in the third part of the article, after which we design a general decentralized predictive protocol based merely on local observations.

11.1.1 Predictive Mechanisms for Flocks with Leaders

To make a rational analysis of the collective predictive mechanisms of flocks with leaders, we consider a typical complex network model, i.e. the WS small-world network model, and use it as the starting framework for the design of such mechanisms. Then, to demonstrate the remarkable improvements endowed by the newly introduced predictive capability, both velocity synchronization and position cohesion performances are evaluated for two dominating flocking models, the A/R and Vicsek models.

As shown in [19], for flocks with a single leader, it is advantageous to consider a small-world-type network obtained by randomly adding long-range connections from the position of the leader predicted several steps ahead to the current position of a few distant agents called *pseudo-leaders*. Other non-special agents are called *followers*. Thus, there are three different kinds of agents: leader (L), pseudo-leaders (P), and followers (F).

11.1.2 Predictive Mechanisms for Flocks Governed by A/R Model

In this scenario, the flock is assumed to move in an m -dimensional space, the standard A/R function [6, 12]

$$G(d_{pL}) = -d_{pL} \left(a - b \cdot \exp(-\|d_{pL}\|_2^2/c) \right) \quad (11.1)$$

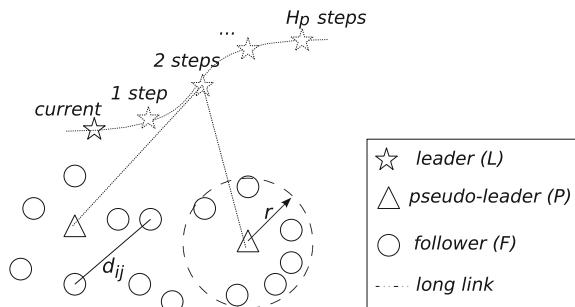
is added to govern the dynamics of the long-range interaction from the leader (L) to each pseudo-leader (P). a, b, c denote free parameters, d_{pL} is an m -dimensional vector pointing from the predicted location of leader L to the current location of a pseudo-leader p , and $\|d_{pL}\|_2$ denotes the Euclidian distance between them, here $\|y\|_2 = \sqrt{y^T y}$. For simplicity, in our model, the motion of the leader is known in advance, and will not be affected by any other agents. Additionally, we assume that each pseudo-leader is able to accurately predict the leader's position H_p steps ahead.

On the other hand, a weaker A/R function, governing the dynamics of the short-range interaction between two arbitrary neighboring agents i and j is given by:

$$g(d_{ij}) = -d_{ij} \left(\tilde{a} - \tilde{b} \cdot \exp(-\|d_{ij}\|_2^2/\tilde{c}) \right), \quad (11.2)$$

where d_{ij} is an m -dimensional vector pointing from the individuals j to i , $\|d_{ij}\|_2 = \sqrt{d_{ij}^T d_{ij}}$ denotes the Euclidean distance between them, and the parameters \tilde{a}, \tilde{b} and \tilde{c} are typically much smaller than a, b , and c , respectively. We denote r as the radius of the influence circle of each agent (see Fig. 11.1). Any two agents of the type F-F, P-P and L-F which are within a distance r of each other are connected by a link whose strength is characterized by the A/R function in Eq. (11.2). Conversely, irrespective of their relative distance, L-P agents are always connected by a link whose strength is described by the A/R function in Eq. (11.1). Note that the leader is never influenced by other agents. To decrease the prediction cost, prediction is only allowed for the L-P interactions. Bearing in mind the physical meaning of A/R functions [12], the positions of a pseudo-leader x_p and a follower x_i is determined by

Fig. 11.1 Predictive mechanism in small-world networks



$$x_p(k+1) = x_p(k) + \underbrace{G(d_{pL}(k+H_p))}_{\text{long link to the leader}} + \underbrace{\sum_{j \neq L, d_{pj}(k) \leq r} g(d_{pj}(k))}_{\text{neighboring links}}, \quad (11.3)$$

and

$$x_i(k+1) = x_i(k) + \underbrace{\sum_{j, d_{ij}(k) \leq r} g(d_{ij}(k))}_{\text{neighboring links}}, \quad (11.4)$$

respectively.

In these equations, k denotes the current discrete-time instant, and $d_{pL}(k+H_p)$ represents the m -dimensional vector pointing from the leader's position H_p steps ahead to the current position of a pseudo-leader.

Now two important questions arise: How many pseudo-leaders are required to yield a satisfactory group behavior? And how many steps should be predicted by each pseudo-leader? In order to extract the number of pseudo-leaders N_{pl} and the role of the prediction horizon H_p , we study their influences on the velocity synchronization and position cohesion performance indexes J_v and J_p which are respectively defined as:

$$J_v = \frac{1}{N-1} \sum_{i=1, i \neq L}^N \|v_i - v_L\|_2, \quad (11.5)$$

$$J_p = \frac{1}{N-1} \sum_{i=1, i \neq L}^N \|d_{iL}\|_2. \quad (11.6)$$

J_v measures the velocity synchronization or formation performance of the flock, where v_L and v_i denote the velocity vectors of the leader and the i th agent (F or P); J_p measures the cohesion performance of the flock, with d_{iL} denoting the distance between agent i (F or P) and the leader.

In Fig. 11.2a, we consider H_p as a fixed parameter and display the curves of J_v with increasing N_{pl} , while Fig. 11.2b, on the contrary, displays the curves of J_v with increasing H_p and fixed N_{pl} . It can be seen from Fig. 11.2a that the curves decrease sharply at the beginning and reach a minimum N_{pl}^* before increasing more slowly. The presence of a minimum at low values of N_{pl} implies that adding just a few pseudo-leaders to the system, which transforms the flock topology from a strongly localized network into a small-world one, will improve the flocking performance optimally in terms of N_{pl} . If more pseudo-leaders than the optimal value N_{pl}^* are added, the flock formation performance starts to worsen as these extra pseudo-leaders become redundant. On the other hand, we also see that increasing H_p can help in reducing J_v in two ways: (i) it decreases J_v for a fixed value of N_{pl} ; (ii) it reduces the optimal value N_{pl}^* . Compared with Fig. 11.2a, the J_v curves in Fig. 11.2b also possess a clear minimum. The presence of this minimum implies that the flock

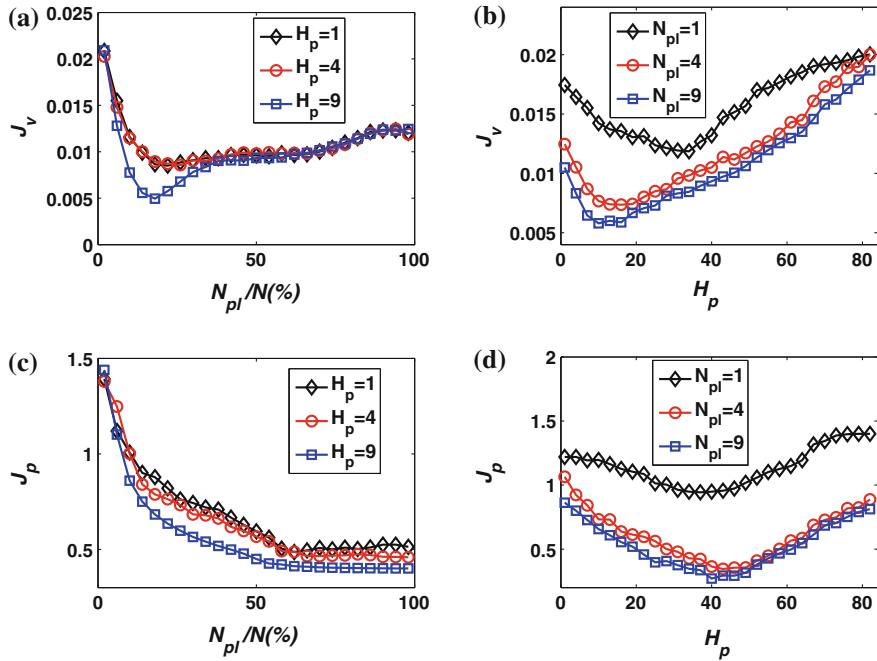


Fig. 11.2 (Color online) The roles of the pseudo-leaders' number N_{pl} (figure **a** and **c**) and prediction horizon H_p (figure **b** and **d**) on a flock with a total of $N = 50$ agents. The leader and the pseudo-leaders are selected randomly among these N agents. Each point is an average over 1000 independent runs. The parameters of the A/R functions (11.1) and (11.2) are $a = 8$, $b = 17.6$, $c = 0.4$, and $\tilde{a} = 1$, $\tilde{b} = 2.2$, $\tilde{c} = 0.2$, respectively. The radius of the influence circle is $r = 0.65$. Each agent starts from a position randomly selected in the square $[0, 1] \times [0, 1]$. Without loss of generality, the trajectory of the leader is set along the curve defined by $x_2 = \sqrt{x_1}$, and the velocity of the leader is $v_{Lx_1}(k) = 0.02$, $v_{Lx_2}(k) = \sqrt{0.02(k+1)} - \sqrt{0.02}k$

formation performance can be remarkably improved with proper predictive capability. Nevertheless, too much vision into the future, namely over-prediction, will worsen the flocking performance as measured by J_v .

Next, we investigate the effects of H_p and N_{pl} on the other important flock performance index J_p . It can be seen from Fig. 11.2c that the curves fall sharply at the beginning and asymptotically approach a stable value as $N_{pl}/N \rightarrow 1$. Contrary to Fig. 11.2a, no clear minimum exists in Fig. 11.2c. As illustrated in Fig. 11.2c, increasing H_p yields smaller values of J_p . Compared with Fig. 11.2c, the curves of Fig. 11.2d do possess a minimum, which is reached at a fairly large H_p . As a consequence, the compactness of the flock is clearly dependent on H_p and over-prediction can be detrimental to it. Furthermore, increasing the number of pseudo-leaders can improve the cohesive flocking performance. Actually, when N_{pl} exceeds a certain value, J_p decreases so slowly that almost no benefit can be gained by further increasing N_{pl} . In brief, suitable insight into the future and moderate number of pseudo-leaders are preferred.

Interestingly, compared with Couzin's results [5], we have shown through the numerical simulations presented in Fig. 11.2 that the incorporation of predictive intelligence can even further reduce the required number of pseudo-leaders.

More significantly, as shown in Fig. 11.2, to achieve a fixed flocking performance, larger predictive capability (larger values of H_p) can compensate for the insufficiency of the pseudo-leaders (N_{pl}) and vice versa. This rule is useful for industrial applications since one can optimize the performance of the dynamic flock by evaluating the costs of the increased predictive capability and the addition of the long-range connections.

11.1.3 Predictive Mechanisms for Flocks Governed by the Vicsek Model

The role of predictive mechanisms highlighted in Sect. 11.1.2 is not merely confined to A/R flocks but quite general. To verify this, we now incorporate the predictive mechanism proposed for small-world interaction patterns in the previous section into another well-accepted flocking model, i.e. the Vicsek model [17], and compare the synchronization performance of the predictive small-world Vicsek model with the one of a classical Vicsek model.

In this model, the velocities v_i of the N agents composing the group are determined simultaneously at each discrete-time instant, and the position of the i th agent is updated according to

$$x_i(k+1) = x_i(k) + v_i(k),$$

where $v_i(k)$ denotes the velocity vector of agent i at time k . For each agent, the velocity vector $v_i(k)$ is characterized by a constant magnitude v and by a direction $\theta_i(k)$ whose dynamics is given by

$$\theta_i(k+1) = \langle \theta_i(k) \rangle_r + \Delta\theta_i,$$

where $\langle \theta_i(k) \rangle_r$ denotes the average direction of all the agents' velocity vectors within a circle of radius r centered on agent i , i.e.,

$$\langle \theta_i(k) \rangle_r = \begin{cases} \arctan \left[\langle \sin(\theta_i(k)) \rangle_r / \langle \cos(\theta_i(k)) \rangle_r \right] \\ \quad \text{if } \langle \cos(\theta_i(k)) \rangle_r \geq 0; \\ \arctan \left[\langle \sin(\theta_i(k)) \rangle_r / \langle \cos(\theta_i(k)) \rangle_r \right] + \pi \\ \quad \text{otherwise,} \end{cases}$$

where $\langle \sin(\theta_i(k)) \rangle_r$ and $\langle \cos(\theta_i(k)) \rangle_r$ denote the average sine and cosine values, and $\Delta\theta_i$ represents a random noise obeying a uniform distribution in the interval $[-\eta/2, \eta/2]$.

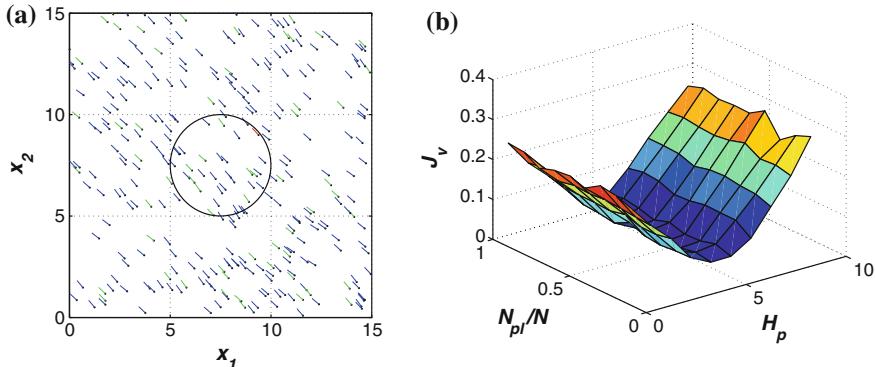


Fig. 11.3 (Color online) **a** Snapshot of the predictive Vicsek flock at the 12th running step. The red particle denotes the leader; the green particles represent the pseudo-leaders and the blue particles denote followers. The centered black circle outlines the trajectory of the leader. For these simulations, the prediction horizon is $H_p = 4$. **b** Velocity synchronization index J_v as a function of the parameters H_p and N_{pl}/N . The parameters of this simulation have been chosen to be $L = 15$, $\eta = 0.1$, $v = 0.15$, $N = 300$, $R = L/6$, and each point is an average over 100 independent runs

As shown in Fig. 11.3a the particles are distributed in a square of dimension $[0, L] \times [0, L]$, and the trajectory of the leader is a circle centered at $(L/2, L/2)$ with radius $R = L/6$ so that the direction of the leader changes constantly. The small-world predictive connection framework shown in Fig. 11.1 is used together with the Vicsek model. Hence, there are always N_{pl} individuals having long-range predictive interactions with the predicted motion of the leader H_p steps ahead. It is shown in Fig. 11.3b that drastic improvement of the velocity synchronization performance can be achieved with moderate prediction horizons. Similar to the case of the A/R model shown in Sect. 11.1.2, one can also conclude that suitable insight into the future and moderate number of pseudo-leaders is preferable.

11.1.4 Decentralized Predictive Mechanism for Flocks Without Leaders

11.1.4.1 Preliminary Concepts

We represent a network of interacting agents by a digraph $G = (\mathcal{V}, \mathcal{E}, A)$, where $\mathcal{V} = \{v_1, \dots, v_N\}$ is the set of *nodes* representing the agents, $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ is the set of *edges*, and A is the $N \times N$ *adjacency matrix* with $a_{ij} \geq 0$ denoting the edge weight from node i to node j . No self-cycle is allowed, hence $a_{ii} = 0$ for all i . The *Laplacian matrix* L is defined as $l_{ii} = \sum_{l \neq i}^N a_{il}$ and $l_{ij} = -a_{ij}$, $\forall i \neq j$. We denote $d_{\max} = \max_i (l_{ii})$ as the *maximum out-degree* of G . A digraph G is called *balanced* if $\forall i$, $\sum_j a_{ij} = \sum_j a_{ji}$. If any two distinct nodes can be connected via

a path that follows the direction of the edges of the digraph, this network is called *strongly connected*. $x_i(t) \in \mathbb{R}$ denotes the *state* of node i , which could represent a certain physical quantity such as attitude, position, temperature, voltage, and so on. Generally, a network is said to have reached *consensus* (or agreement) if and only if $x_i = x_j$ for all $i, j \in \mathcal{V}$. Whenever the nodes of a network are all in agreement, their common value is called the *group decision value*. If this value is $\bar{x}(0) = (\sum_{i=1}^N x_i(0))/N$, the network is said to have reached the *average-consensus*.

11.1.4.2 Prediction Based on Local Information

We consider the linear discrete-time network model:

$$x(k+1) = P_\varepsilon x(k), \quad (11.7)$$

based on which the m -step-ahead future state of an individual i can easily be derived:

$$x_i(k+m) = e_i P_\varepsilon^m x(k), \quad (11.8)$$

where $e_i = [0, \dots, 0, 1^{ith}, 0, \dots, 0]$. In the following paragraphs, we show that, even if such global information is unavailable to each node of the network (which is typically the case in natural bio-groups and engineering multi-agent systems), each individual may still be capable of estimating its own and its neighbors' future states by using the present and past information it collects from its neighbors.

More precisely, let $Z_i(k) = [z_{1,i}^T(k), \dots, z_{N,i}^T(k)]^T$ denote the historical state sequence of length N for the i th individual with $z_{l,i}(k) \triangleq x_i(k+1-l), l = 1, \dots, N$. Using (11.7), it is easy to show that the following relations exist

$$x(k-N) = \Phi_i^{-1} Z_i(k), \quad (11.9)$$

and

$$x_j(k+1) = B_{j,i} Z_i(k), \quad j = 1, \dots, N \quad (11.10)$$

with

$$\Phi_i \triangleq \left[(e_i P_\varepsilon^N)^T, \dots, (e_i P_\varepsilon)^T \right]^T \quad (11.11)$$

and

$$B_{j,i} \triangleq e_j P_\varepsilon^{N+1} \Phi_i^{-1}, \quad (11.12)$$

provided that Φ_i is invertible.

Analogously, the future dynamics of individual i and its neighbors $j \in \mathcal{N}(i)$ ¹ can be iterated as follows

$$Z_i(k+m) = A_i^m Z_i(k), \quad (11.13)$$

$$x_j(k+m) = B_{j,i} A_i^{m-1} Z_i(k), \quad (11.14)$$

where

$$A_i \triangleq \begin{bmatrix} B_{i,i} \\ I_{N-1} \mathbf{0}_{(N-1) \times 1} \end{bmatrix} \quad (11.15)$$

$$j \in \mathcal{N}(i) \cup i; m = 1, \dots, H_p,$$

and H_p is the prediction horizon. Therefore, for an arbitrary individual i , provided that the constant parameters $B_{j,i}$ ($j \in \mathcal{N}(i) \cup i$) (see Eqs. (11.12) and (11.15)) can be appropriately estimated, the future states of itself and its neighbors can be effectively predicted solely using the historical local state sequences $Z_j(k)$ ($j \in \mathcal{N}(i) \cup i$) observable by individual i .

Estimation of $B_{j,i}$ can be realized provided that each individual possesses a memory of sufficient capacity, allowing it to store the length- N historical state sequences of $x_i(k)$ and $x_j(k)$. More precisely, since at time step k , individual i has already stored $x_j(k-n)$ and $Z_i(k-n-1)$ ($j \in \mathcal{N}(i) \cup i$, $n = 0, 1, \dots, N-1$) into its memory, we can use

$$x_j(k-n) = B_{j,i} Z_i(k-n-1), \quad (11.16)$$

and perform a least square estimation (LSE) [9] to obtain the estimates $\hat{B}_{j,i}$ of the row vectors $B_{j,i}$.

Furthermore, in natural flocks/swarms, it frequently happens that some individuals escape the neighborhood of an individual and enter the neighborhood of others, or that some new individuals join the group while some others leave it, making the network topology change continuously. Fortunately, if the topology modification rate is not too high, the proposed decentralized prediction remains feasible. Indeed, since the prediction is based on the historical information sequence of the last $2N$ steps before current time k (see Eq. (11.16)), and is aimed at the prediction of the future H_p steps (see Eq. (11.14)), the decentralized predictive protocol (11.16) remains valid provided that the topology remains constant during $2N + H_p$ steps. Thus, the upper bound of the topology changing rate is $1/(2N + H_p)$.

The decentralized predictive protocol (11.16) is illustrated on the 5-node network shown in Fig. 11.4, in which the topology is changed from Fig. 11.4a to b at the 47th step. Numerical simulations show that, before the topology switch the prediction error $e_{p,i}(k)$ remains small (less than 10^{-7}). Note that the prediction error of node

¹ $\mathcal{N}(i)$ denotes the set of neighbors of i . More precisely, we say that a node j is a neighbor of a node i , denoted by $j \in \mathcal{N}(i)$ if, and only if, the corresponding element of the associated adjacency matrix $a_{ij} \neq 0$.

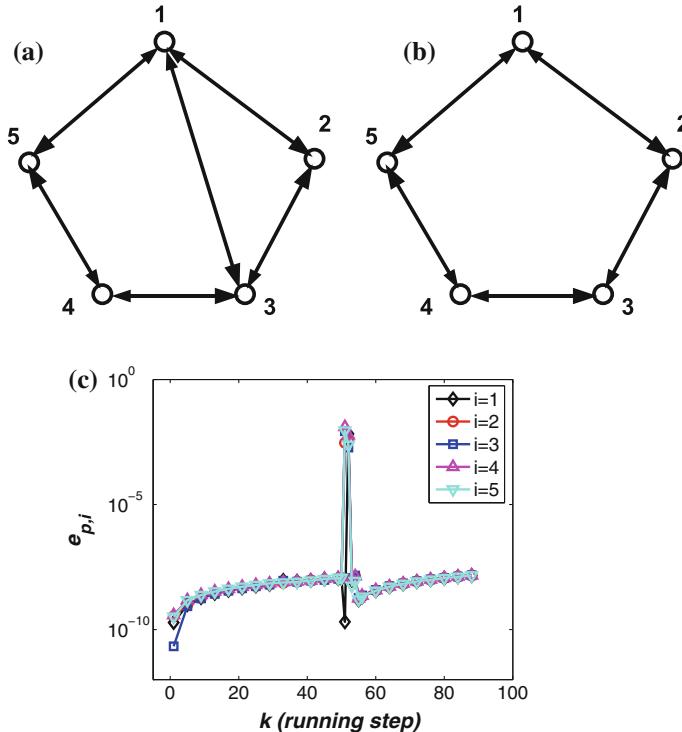


Fig. 11.4 (Color online) The network topology is switched from **a** to **b** at the 47th running step; **c** gives the instantaneous prediction error $e_{p,i}(k)$. Here, $\varepsilon = 0.35$, $a_{ij} = 1$ if $(i, j) \in \mathcal{E}$, and $a_{ij} = 0$ otherwise, the initial state of each node is selected randomly in the range $[0, 150]$, and each point is an average over 100 independent runs

i is defined as $e_{p,i}(k + m) = \|x_i(k + m) - \hat{x}_i(k + m)\|_2$ ($m = 1, \dots, H_p$) with $\hat{x}_i(k + m) = e_i \hat{A}_i^m Z_i(k)$ and \hat{A}_i being the least square estimation of A_i given in Eq. (11.15). When the topology switches, the prediction error $e_{p,i}(k)$ rises sharply to approach 0.1 and then begins to oscillate as a result of the transient adaptive process of the prediction. After less than N steps, $e_{p,i}(k)$ settles down to a level lower than 10^{-7} . In this way, both the prediction accuracy and the adaptive capability for topology variations have been illustrated through simulation.

11.1.4.3 Decentralized Predictive Protocol

We add a prediction control input to the routine dynamics given in (11.13) for $m = 1$:

$$Z_i(k + 1) = A_i Z_i(k) + e_1^T v_i(k), \quad (11.17)$$

$$x_i(k + 1) = e_1 A_i Z_i(k) + v_i(k), \quad (11.18)$$

$$x_j(k+1) = B_{j,i}Z_i(k), \quad (11.19)$$

where $v_i(k)$ is an additional term representing the MPC action. Using the predictive protocol (11.17)–(11.19), the future states of agent i can be predicted based on the currently available historical state sequence $Z_i(k)$ as follows:

$$X_i(k+1) = P_{Z_i}Z_i(k) + P_{U_i}U_i(k), \quad (11.20)$$

$$X_j(k+1) = P_{Z_j}Z_i(k) + P_{U_j}U_i(k) \quad (11.21)$$

Here, $H_u \leq H_p$ denotes the control horizon, i.e. the length of the predicted control sequence, and $X_i(k+1) = [x_i(k+1), \dots, x_i(k+H_p)]^T$, $X_j(k+1) = [x_j(k+1), \dots, x_j(k+H_p)]^T$, $U_i(k) = [v_i(k), \dots, v_i(k+H_u-1)]^T$, $P_{Z_i} = \left[(e_1 A_i)^T, \dots, (e_1 A_i^{H_p})^T \right]_{H_p \times N}^T$, $P_{Z_j} = \left[B_{j,i}^T, (B_{j,i} A_i)^T, \dots, (B_{j,i} A_i^{H_p-1})^T \right]_{H_p \times N}^T$,

$$P_{U_i} = \begin{bmatrix} 1 & & & & \\ e_1 A_i e_1^T & 1 & & & \\ \vdots & & \ddots & & \\ e_1 A_i^{H_u-1} e_1^T & \dots & \dots & 1 & \\ e_1 A_i^{H_u} e_1^T & \dots & e_1 A_i^2 e_1^T & e_1 A_i e_1^T + 1 & \\ \vdots & & \vdots & & \vdots \\ e_1 A_i^{H_p-1} e_1^T & \dots & e_1 A_i^{H_p-H_u+1} e_1^T & e_1 A_i^{H_p-H_u} e_1^T & \\ & & & e_1 A_i^{H_p-H_u-1} e_1^T & \\ & & & + \dots + 1 & \end{bmatrix},$$

and

$$P_{U_j} = \begin{bmatrix} 0 & & & & \\ B_{j,i} e_1^T & 0 & & & \\ \vdots & & \ddots & & \\ B_{j,i} A_i^{H_u-2} e_1^T & \dots & \dots & 0 & \\ B_{j,i} A_i^{H_u-1} e_1^T & \dots & B_{j,i} A_i e_1^T & B_{j,i} e_1^T & \\ \vdots & & \vdots & & \vdots \\ B_{j,i} A_i^{H_p-2} e_1^T & \dots & B_{j,i} A_i^{H_p-H_u} e_1^T & B_{j,i} A_i^{H_p-H_u-1} e_1^T & \\ & & & + B_{j,i} A_i^{H_p-H_u-2} e_1^T & \\ & & & + \dots + B_{j,i} e_1^T & \end{bmatrix}.$$

We represent the state difference between individual i and its neighbors by a vector $\Delta \mathbb{X}_i(k+1) = \text{row} \{ \Delta X_{r,s}(k+1) | r, s \in \mathcal{N}(i) \cup i \text{ and } r > s \}$ where $\Delta X_{r,s}(k+1) = X_r(k+1) - X_s(k+1)$ and the notation $y = \text{row}\{y_i\}$ ($i = 1, \dots, n$) means that y is composed of rows of y_i , i.e., $y = \text{row}\{y_i\} \Leftrightarrow y = [y_1^T, \dots, y_n^T]^T$. For instance, as shown in Fig. 11.4b, individual 2 has two neighbors (numbered 1 and 3), therefore

$$\Delta \mathbb{X}_2(k+1) = \left[(X_2(k+1) - X_1(k+1))^T, (X_3(k+1) - X_2(k+1))^T, (X_3(k+1) - X_1(k+1))^T \right]^T.$$

It then follows from Eqs. (11.20) and (11.21) that

$$\Delta \mathbb{X}_i(k+1) = P_{\mathbb{Z}_i} Z_i(k) + P_{\mathbb{U}_i} U_i(k) \quad (11.22)$$

with $r, s \in \mathcal{N}(i) \cup i$ and $r > s$, $P_{\mathbb{Z}_i} = \text{row}\{P_{Z_r} - P_{Z_s}\}$ and $P_{\mathbb{U}_i} = \text{row}\{P_{U_r} - P_{U_s}\}$.

To solve the consensus problem, we first set the moving horizon optimization index that defines the decentralized MPC consensus problem as follows:

$$J_i(k) = \|\Delta \mathbb{X}_i(k+1)\|_{Q_i}^2 + \|U_i(k)\|_{R_i}^2, \quad (11.23)$$

where Q_i and R_i are compatible real, symmetric, positive definite weighting matrices. In general, the weighting matrices can be set as

$$Q_i = q I_{H_p N_i (N_i+1)/2} \quad (q > 0) \quad \text{and} \quad R_i = I_{H_u}, \quad (11.24)$$

where N_i is the number of neighbors of individual i . In the optimization index (11.23), the first term penalizes the disagreement among the neighborhood of agent i over the future H_p steps, while the second penalizes the additional MPC control energy $U_i(k)$. In order to minimize (11.23), we compute $\partial J_i(k)/\partial U_i(k) = 0$, and consequently obtain the optimal MPC action as:

$$v_i(k) = P_{MPC,i} Z_i(k) \quad (11.25)$$

with

$$P_{MPC,i} = -[1, 0, \dots, 0]_{1 \times H_u} (P_{\mathbb{U}_i}^T Q_i P_{\mathbb{U}_i} + R_i)^{-1} P_{\mathbb{U}_i}^T Q_i P_{\mathbb{Z}_i}. \quad (11.26)$$

Note that, in this decentralized MPC, the control law $v_i(k)$ is calculated and implemented by each individual, which is totally different from centralized MPC.

Substituting Eq. (11.25) into Eq. (11.17) yields

$$Z(k+1) = W Z(k) \quad (11.27)$$

with $W = \text{diag}\{A_i + e_1^T P_{MPC,i}\}_{i=1,\dots,N}$ and $Z(k) = [Z_1(k)^T, \dots, Z_N(k)^T]^T$.

Based on the above discussion, the decentralized MPC algorithm can be divided into two stages:

Stage 1 (Pure identification stage): In the first $2N$ steps, the LSE method (11.16) is implemented to yield initial estimates \hat{A}_i and $\hat{B}_{j,i}$ of the neighbors' dynamics;

Stage 2 (identification and control stage): From the $2N + 1$ th step, the MPC is kicked off. At each step k , the matrices $P_{MPC,i}$ are calculated according to Eq. (11.26) using the estimates \hat{A}_i , $\hat{B}_{j,i}$. The MPC term $v(k)$ is then computed according to Eq. (11.25) and introduced into each node as shown in Eq. (11.18). Finally, the LSE method (11.16) is iterated to yield the updated estimates \hat{A}_i and $\hat{B}_{j,i}$.

To support the above proposed decentralized MPC, it is necessary to give some sufficient conditions guaranteeing convergence towards average-consensus. To this end, we first provide a lemma on average-consensus.

Lemma 11.1 [20] For any matrix $W \in \mathbb{R}^{M \times M}$, the equation

$$\lim_{k \rightarrow \infty} W^k = \mathbf{1}_M \mathbf{1}_M^{T/M} \quad (11.28)$$

with $\mathbf{1}_M = [1, \dots, 1]_{M \times 1}^T$ holds if and only if assumptions A1 and A2 hold:

A1: The following holds for W

$$W\mathbf{1}_M = W^T \mathbf{1}_M = \mathbf{1}_M; \quad (11.29)$$

A2: The matrix W has a simple eigenvalue at 1 and all its other eigenvalues inside the open unit circle.

Based on Lemma 11.1, we give hereafter the necessary and sufficient conditions guaranteeing average-consensus for the proposed decentralized MPC protocol.

Theorem 11.1 For an N -node balanced network with dynamics determined by (11.27), provided that Φ_i , $i = 1, \dots, N$ (see Eq. (11.11)) are invertible and $\hat{B}_{j,i} = B_{j,i}$ (see Eq. (11.12)), then the system state $x(k)$ asymptotically converges to the average-consensus value $\bar{x}(0)\mathbf{1}_N$ with $\bar{x}(0) \triangleq 1/N \sum_{i=1}^N x_i(0)$ if and only if assumptions A3 and A2 hold:

A3: The following equality holds

$$\mathbf{1}_N^T \cdot (A_i + e_1^T P_{MPC,i}) = \mathbf{1}_N^T \quad (i = 1, \dots, N). \quad (11.30)$$

Furthermore, the eigenvalue distribution of W (see Eq. (11.27)) is always much smaller and closer to the origin than the one of P_ε , which explains the overall higher consensus speed of the decentralized MPC protocol. More significantly, when ε is increased beyond the threshold $1/d_{\max}$ (see Fig. 11.6b where $\varepsilon = 2$), some of the eigenvalues of P_ε start escaping the unit circle, making the disagreement function diverge, whereas all the eigenvalues of W remain inside the unit circle (except one that is always located at 1), which ensures its convergence. Clearly, from the mathematical point of view, the effect of predictive mechanism is to drive the escaping eigenvalues towards the origin. One may notice that, compared with nominal MPC, the eigenvalue distribution of decentralized MPC is less compact around the origin. This is due to the parameter estimation error which inevitably leads to slight inaccuracies in the state predictions.

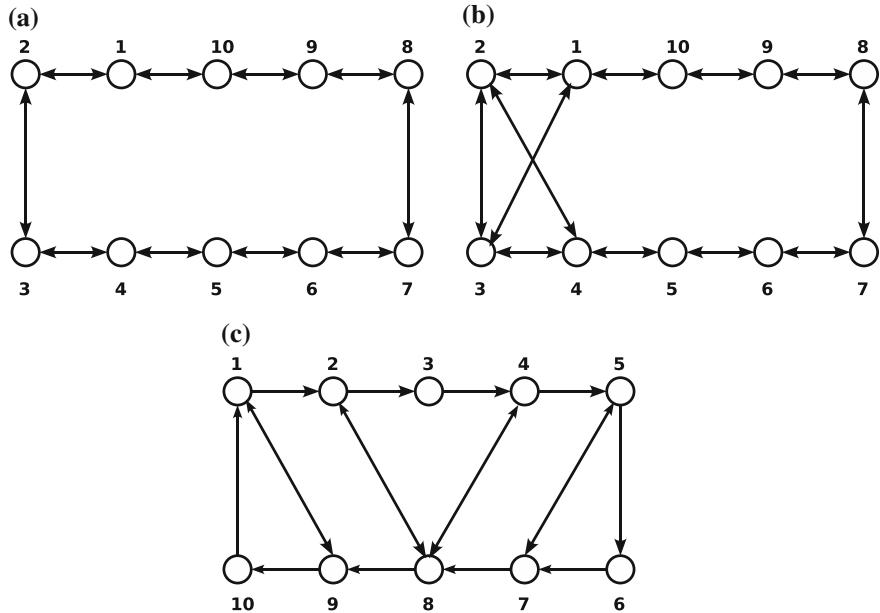


Fig. 11.5 Network topologies used

To vividly illustrate the virtues of the decentralized predictive consensus protocol, we present some simulation results comparing the convergence speeds obtained using the routine protocol given in Eq. (11.7) and the proposed predictive protocol given in Eq. (11.27) on the 10-node network given in Fig. 11.5a. Once again, since the objective is to reach average-consensus, the instantaneous disagreement index is set as $D(k) = \|x(k) - \mathbf{1}_N \bar{x}(0)\|_2^2$.

In the convergent case of the routine protocol, e.g. for $\varepsilon \leq 1/d_{\max}$, the decentralized MPC consensus protocol (11.27) yields an increase in the convergence speed towards average-consensus (by a factor of 3 approximately), as shown in Fig. 11.7a. Furthermore, even when the routine convergence conditions are violated, e.g. $\varepsilon > 1/d_{\max}$, it is observed in Fig. 11.7b, c that the decentralized MPC consensus protocol still allows asymptotic convergence to average-consensus. Thus, the range of sampling period ε leading to asymptotic convergence towards consensus is expanded using the decentralized predictive mechanism. Interestingly, one may notice the slight oscillations of the MPC consensus protocol's $D(k)$ curve in Fig. 11.7a, b, which root in the online identification and adaptation processes performed by each node.

Finally, to demonstrate the decentralized MPC's superiority in handling network topology switches, we compare the consensus performances of the MPC and routine protocols in Fig. 11.8, when the topology is switched from Fig. 11.5a, b at the 43rd running step. For the routine protocol, since $\{0.48, 0.49\} \in [1/3, 1/2] (1/2 \text{ and } 1/3)$

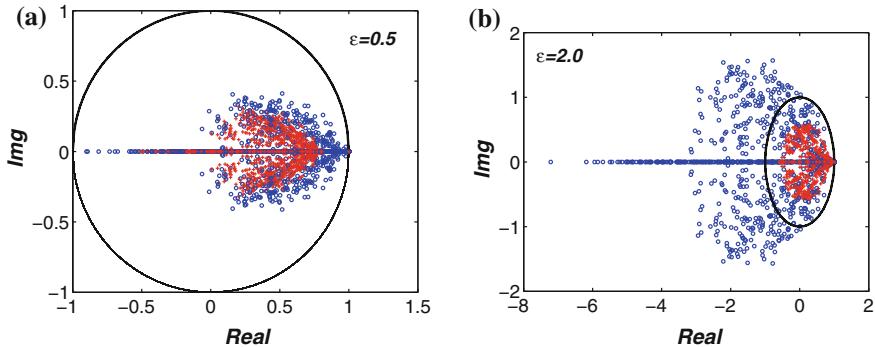


Fig. 11.6 (Color online) Eigenvalue distributions for different ε . Blue (\circ) and red (+) represent the eigenvalues of P_ε and W (see Eq. (11.27)) over 100 runs, respectively. The simulations are implemented on the balanced network with topology given in Fig. 11.5c. The eigenvalue distribution of W correspond to snapshots at the 50th running step. The black circle denotes the unit circle in the complex plane. Here, $H_p = 7$, $H_u = 1$, $q = 0.015$, and each entry l_{ij} ($j \neq i$, and $(i, j) \in \mathcal{E}$) is chosen randomly in $[-1, 0]$ such that the resulting network is balanced. The associated values of d_{\max} lie in $[0.9, 2.7]$

are the $1/d_{\max}$ values of the topologies considered in Fig. 11.4a, b, respectively), the average-consensus tendency is broken shortly after the network topology switch. On the contrary, the average-consensus tendency holds all along for the decentralized MPC consensus protocol since it can adapt to the topology changes and use the additional MPC term (see Eq. (11.18)) to steer each node towards its neighbors.

11.1.5 Conclusion

To extract the role of predictive mechanisms that may extensively exist in abundant natural bio-groups, we surveyed some recent investigations on collective group behaviors, and designed predictive mechanisms for flocks with and without leaders. By using mathematical analysis and numerical simulations, we illustrated the advantages of such predictive protocols for both the exploration of emergent behaviors and the design of autonomous and reliable consensus networks.

For natural science, the contribution of this work lies in its ability to explain why individuals of biological flocks/swarms like fireflies and deep-sea fish do not communicate very frequently all along but just now and then during the whole dynamic process. From the industrial application point of view, the value of this work is two-fold. The consensus performance is significantly enhanced while the communication energy or cost is effectively reduced. All these virtues are only at the cost of giving the agents the capabilities of storing past states and making predictions. This work

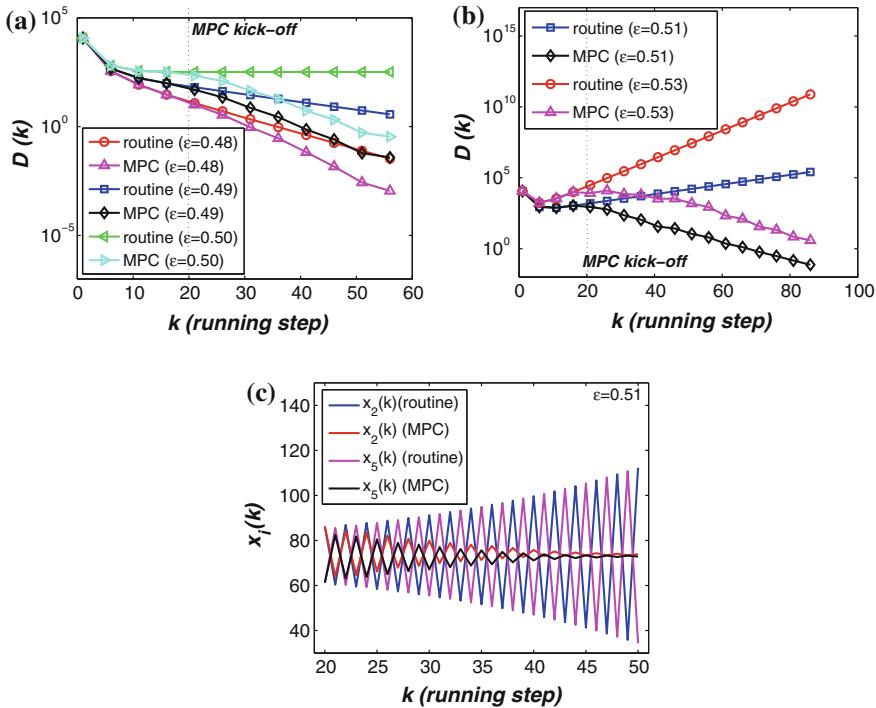


Fig. 11.7 (Color online) Consensus performance comparison of the routine and predictive protocols on the balanced network represented in Fig. 11.5a for which $d_{\max} = 2$. **a** Time evolution of the instantaneous disagreement $D(k)$ for $\varepsilon \leq 1/d_{\max}$ (convergent case of the routine protocol); **b** Time evolution of the instantaneous disagreement $D(k)$ for $\varepsilon > 1/d_{\max}$ (divergent case of the routine protocol); **c** state evolution for $\varepsilon = 0.51$. Here, $a_{ij} = 1$ if $(i, j) \in \mathcal{E}$, otherwise $a_{ij} = 0$, the initial state $x_i(0)$ is selected randomly in $[0, 150]$, $H_p = 7$, $H_u = 1$, $N = 10$, $q = 0.015$. Each point is an average over 500 independent runs

is just a first attempt to reveal the role of collective predictive mechanism for natural flocks/swarms and thereby improving the performances of industrial multi-agent systems' collective behaviors. We believe that numerous research issues remain open in the ongoing efforts to design predictive protocols that are efficient, robust and scalable to large-sized systems. We also expect the coming years to embrace an intensive development about the collective behavior coordination via predictive mechanism and the relevant applications in multi-agent industrial systems.

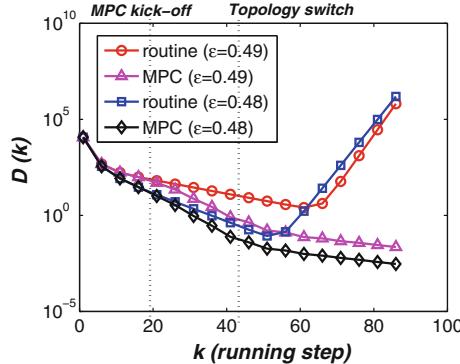


Fig. 11.8 (Color online) Consensus performance with switching topology. The balanced network topology is switched from Fig. 11.5a to b at the 43rd running step. For these simulations, $a_{ij} = 1$ if $(i, j) \in \mathcal{E}$, and $a_{ij} = 0$ otherwise, the initial state $x_i(0)$ is selected randomly in $[0, 150]$, $H_p = 7$, $H_u = 1$, $N = 10$, and $q = 0.015$. Each point is an average over 500 independent runs

11.2 Predictive Consensus Control for Single-Integrator Networks Without Input Constraints

In recent years, more and more scholars have used pinning control to address the synchronization or consensus problems [8, 18]. In this methodology, just a small proportion of the nodes are selected as the pinning nodes as shown in Fig. 11.9, who know the global target of the entire MAS or can achieve the states of all the other nodes. By regulating these few pinning nodes, the synchronization or consensus procedure will be substantially accelerated at a low communication cost [8, 18]. In this section, we propose some innovative solutions based on model predictive control (MPC), which is a widely used approach for its capability of handling the inter-individual coupling and the input/output constraints [10, 11, 14]. Fortunately, this pinning control framework can nicely act as a niche platform for predictive control of MASs, since the pinning nodes can be used to provide an accurate future

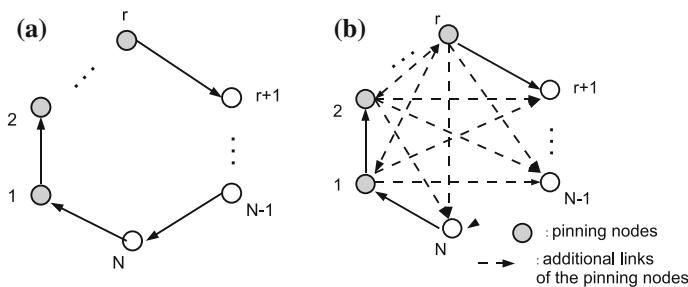


Fig. 11.9 (a) A network before pinning; (b) after pinning. Here, the pinning nodes $r \ll N$

state trajectory prediction due to the availability of the global MASs information at these nodes. In detail, we will show that, by incorporating one kind of predictive mechanism to these pinning nodes, the convergence speed towards consensus will be substantially increased.

11.2.1 Problem Description

We will focus on the digraphs with $r \ll N$ pinning nodes as shown in Fig. 11.9. Note that, the other $N - r$ common nodes do not necessarily have the capability to obtain the state information from the pinning nodes. Here, we define *pinning nodes* as the ones who are always connected to all the other nodes (including the other pinning nodes), i.e., $a_{i,j} > 0$ for $i \in \mathcal{V}_p$, $j \in \mathcal{V}$ and $j \neq i$. Here, \mathcal{V}_p is set of the r pinning nodes. Therefore, pinning nodes are defined to be the ones who can always obtain the state information $x(t)$ from all the other nodes. Without loss of generality, we set $\mathcal{V}_p = \{v_1, \dots, v_r\}$. Of course, the r pinning nodes can also be deemed as “leaders” who always have connections to all the other nodes.

Let $x_i \in \mathbb{R}$ denote the state of node i . Generally, we say that the nodes of a network have reached *consensus* if and only if $x_i = x_j$ for all $v_i, v_j \in \mathcal{V}$ ($i \neq j$) [12]. We will address the consensus speeding up problems concerning networks with single-integrator dynamics, where $r \ll N$.

The dynamics of a network of discrete-time integrator agents is defined by [12]:

$$x(k+1) = P_\varepsilon x(k) \quad (11.31)$$

with $P_\varepsilon = I_N - \varepsilon L$, $L = \{l_{ij}\}_{i,j=1,\dots,N} \in \mathbb{R}^{N \times N}$ denoting the *graph Laplacian matrix* induced by the topology G and being defined as $l_{ii} = \sum_{l \neq i}^N a_{il}$, $\forall i = 1, \dots, N$ and $l_{ij} = -a_{ij}$, $\forall i \neq j$. Here, ε denotes the sampling period or step-size, and $I_N \in \mathbb{R}^{N \times N}$ is the identity matrix of dimension N .

For digraph with $r \ll N$ pinning nodes as shown in Fig. 11.9b, we seek to design a suitable predictive mechanism represented by an additional state matrix P_{PPC} , which will be added to P_ε in (11.31), so as to increase the convergence speed towards consensus. Here, the subscript “PPC” represents “predictive pinning control”. Of course, P_{PPC} only have influence on the dynamics of the pinning nodes of the network, which correspond to the first r rows of P_ε . Apparently, because of the existence of the r pinning nodes, the network considered in the rest always has a rooted directed spanning tree over time [15], and hence the consensus is naturally guaranteed.

11.2.2 PPC of Single-Integrator Networks

In order to improve the consensus performances, we replace the classical control protocol $u_i(k) = -\sum_{j=1}^N a_{ij} \Delta x_{i,j}(k)$ [12] by the following PPC consensus protocol:

$$u_i(k) = -\sum_{j=1}^N a_{ij} \Delta x_{i,j}(k) + v_i(k), \quad i \in \mathcal{V}_p \subset \mathcal{V}, \quad (11.32)$$

$$u_i(k) = -\sum_{j=1}^N a_{ij} \Delta x_{i,j}(k), \quad i \notin \mathcal{V}_p, \quad (11.33)$$

where $v_i(k)$ is an additional term representing the PPC action, and $\Delta x_{i,j}(k) = x_i(k) - x_j(k)$. Recall that $\mathcal{V}_p = \{v_1, \dots, v_r \mid r \ll N\}$ are the pinning nodes, and that we seek to accelerate the consensus procedure by incorporating a predictive mechanism into \mathcal{V}_p , which always know the states of all the other nodes.

With this PPC protocol, the network dynamics are given by

$$x(k+1) = P_\varepsilon x(k) + \psi v(k) \quad (11.34)$$

with $v(k) = [v_1(k), \dots, v_r(k)]^T$ representing the PPC decision values for the pinning nodes \mathcal{V}_p and $\psi = [I_{r \times r}, \mathbf{0}_{r \times (N-r)}]^T$. The PPC element $v(k)$ will be calculated by solving a specific receding-horizon optimization problem as described below.

Using the consensus protocol (11.34), the future network state can be predicted based on the current state value $x(k)$ as follows:

$$\begin{aligned} x(k+2) &= P_\varepsilon^2 x(k) + P_\varepsilon \psi v(k) + \psi v(k+1), \\ &\quad \vdots \\ x(k+H_u) &= P_\varepsilon^{H_u} x(k) + \sum_{j=0}^{H_u-1} (P_\varepsilon^{H_u-j-1} \psi v(k+j)), \\ x(k+H_u+1) &= P_\varepsilon^{H_u+1} x(k) + \sum_{j=0}^{H_u-2} (P_\varepsilon^{H_u-j} \psi v(k+j)) \\ &\quad + (P_\varepsilon + I) \psi v(k+H_u-1), \\ &\quad \vdots \\ x(k+H_p) &= P_\varepsilon^{H_p} x(k) + \sum_{j=0}^{H_u-2} (P_\varepsilon^{H_p-j-1} \psi v(k+j)) \\ &\quad + \sum_{j=0}^{H_p-H_u} P_\varepsilon^j \psi v(k+H_u-1). \end{aligned}$$

Here, the integers H_p and H_u represent the prediction and control horizons, respectively.

In this way, the future evolution of the network can be predicted H_p steps ahead, as

$$X(k+1) = P_X x(k) + P_U U(k) \quad (11.35)$$

with

$$\begin{aligned} X^T(k+1) &= [x^T(k+1), \dots, x^T(k+H_p)] \in \mathbb{R}^{1 \times H_p N}, \\ U^T(k) &= [v^T(k), \dots, v^T(k+H_u-1)] \in \mathbb{R}^{1 \times H_u r}, \end{aligned}$$

$P_X^T = \left[P_\varepsilon^T, \dots, (P_\varepsilon^{H_p})^T \right] \in \mathbb{R}^{N \times H_p N}$, and the matrix $P_U \in \mathbb{R}^{H_p N \times H_u r}$ has the following structure

$$P_U = \left[\begin{array}{cccccc} I_N & & & & & \\ P_\varepsilon & I_N & & & & \\ \vdots & & \ddots & & & \\ P_\varepsilon^{H_u-1} & P_\varepsilon^{H_u-2} & \dots & & I_N & \\ P_\varepsilon^{H_u} & \dots & P_\varepsilon^2 & & P_\varepsilon + I_N & \\ \vdots & & \vdots & & \vdots & \\ P_\varepsilon^{H_p-1} & \dots & P_\varepsilon^{H_p-H_u+1} & P_\varepsilon^{H_p-H_u} & & \\ & & & + P_\varepsilon^{H_p-H_u-1} & & \\ & & & & + \dots + I_N & \end{array} \right] \times \Psi \quad (11.36)$$

with $\Psi = \text{diag}\{\psi, \dots, \psi\}_{H_u N \times H_u r}$ and ψ given in Eq.(11.34).

Bearing in mind the goal of consensus, i.e., eliminating the disagreement of all the individuals of the network, we first calculate the state derivative of agents i and j in the network, $m \in \{1, \dots, H_p\}$ steps ahead, using the operator

$$\Delta x_{i,j}(k+m) := x_i(k+m) - x_j(k+m) = e_{i,j}x(k+m) \quad (11.37)$$

with $e_{i,j} := e_i - e_j \in \mathbb{R}^{1 \times N}$ and $e_j := [0, \dots, 0, 1_{j\text{th}}, 0, \dots, 0]$ where only the j th element is non-zero. Based on (11.37), the network state derivative vector m steps ahead can be defined by

$$\Delta x(k+m) := [\Delta x_{1,2}(k+m), \dots, \Delta x_{1,N}(k+m), \Delta x_{2,3}(k+m), \dots, \Delta x_{2,N}(k+m), \dots, \Delta x_{N-1,N}(k+m)]^T \in \mathbb{R}^{N(N-1)/2 \times 1}.$$

Consequently, the future evolution of the network's state derivative can be predicted H_p steps ahead as follows:

$$\begin{aligned} \Delta x(k+1) &= ex(k+1), \\ &\vdots \\ \Delta x(k+H_p) &= ex(k+H_p) \end{aligned} \quad (11.38)$$

with

$$e := [e_{1,2}^T, \dots, e_{1,N}^T, e_{2,3}^T, \dots, e_{2,N}^T, \dots, e_{N-1,N}^T]^T \in \mathbb{R}^{N(N-1)/2 \times N}. \quad (11.39)$$

It then follows from (11.38) that

$$\begin{aligned}\Delta X(k+1) &:= \left[\Delta x(k+1)^T, \dots, \Delta x(k+H_p)^T \right]^T \\ &= EX(k+1) = E(P_X x(k) + P_U U(k)) \\ &= P_{XE} x(k) + P_{UE} U(k) \in \mathbb{R}^{H_p N(N-1)/2 \times 1}\end{aligned}\quad (11.40)$$

with $E := \text{diag}(e, \dots, e) \in \mathbb{R}^{H_p N(N-1)/2 \times H_p N}$, $P_{XE} := EP_X$ and $P_{UE} := EP_U$.

To solve the consensus problem, we first set the receding-horizon optimization index as:

$$J(k) = \|\Delta X(k+1)\|_Q^2 + \|U(k)\|_R^2, \quad (11.41)$$

where Q and R are compatible real, symmetric, positive definite weighting matrices, and $\| * \|_Q^2 = *^T Q *$. For simplicity, the weighting matrices can be set as

$$Q = qI \quad (q > 0) \quad \text{and} \quad R = I. \quad (11.42)$$

In the optimization index (11.41), the first term penalizes the state derivative between each pair of states over the future H_p prediction steps, while the second term penalizes the PPC control energy over the future H_u control steps. In order to minimize (11.41), we compute the values of $U(k)$ that yield $\partial J(k)/\partial U(k) = 0$ to obtain the optimal PPC action by

$$\partial J(k)/\partial U(k) = 2P_{UE}^T Q P_{XE} x(k) + 2(P_{UE}^T Q P_{UE} + R)U(k) = 0,$$

thus $U(k) = -(P_{UE}^T Q P_{UE} + R)^{-1} P_{UE}^T Q P_{XE} x(k)$, and the first N entries of $U(k)$ are extracted as the optimal PPC action:

$$v(k) = \check{P}_{PPC} x(k), \quad (11.43)$$

where $\check{P}_{PPC} = -[I_r, \mathbf{0}_r, \dots, \mathbf{0}_r]_{r \times H_u r} \cdot (P_{UE}^T Q P_{UE} + R)^{-1} P_{UE}^T Q P_{XE}$. The associated closed-loop dynamics can then be written as

$$x(k+1) = (P_\varepsilon + P_{PPC})x(k) \quad (11.44)$$

with $P_{PPC} = \psi \check{P}_{PPC}$.

For conciseness, we hereafter give a new definition on matrix spectrum.

Definition 11.1

$$\rho_m(D) := \rho(D - B_m B_m^T / N), \quad (11.45)$$

where $\rho(\cdot)$ denote the matrix spectrum radius [7].

We are ready to propose the main theorem for single-integrator networks in Eq.(11.44) as follows so as to demonstrate the merits of the proposed PPC consensus protocol.

Theorem 11.2 (Spectrum strict compression theorem–single) *Consider an N -node network $G = (\mathcal{V}, \mathcal{E}, A)$ whose dynamics are described by (11.44), and with the associated weighting matrices given by (11.42). One then has*

$$\rho(P_\varepsilon + P_{PPC} - \mathbf{1}\mathbf{1}^T/N) < \rho(P_\varepsilon - \mathbf{1}\mathbf{1}^T/N). \quad (11.46)$$

Now, it is safe to conclude from Theorem 11.2 that, except for the simple eigenvalue at 1, the spectrum of P_ε is effectively compressed by P_{PPC} , and hence the convergence speed towards consensus is effectively increased [12, 15].

11.2.3 Numerical Simulation

Without loss of generality, we consider a class of ring-shaped digraphs with r pinning nodes as shown in Fig. 11.9. The adjacency matrix A fulfills: (i) $a_{i,j} = 1$ with $i \in \mathcal{V}_p$, $j \in \mathcal{V}$ and $j \neq i$; (ii) $a_{i,i+1} = 1$ with $i = 1, \dots, N-1$; (iii) $a_{N,1} = 1$; and (iv) all the other entries of A are zeros. Since the objective is to reach consensus, the instantaneous disagreement index is typically set as $D(k) := \|x(k) - \mathbf{1}\mu^T \bar{x}(0)\|_2^2$.

Due to the similarity of using a linear quadratic regulator (LQR) to yield an optimal control law (see Eqs.(11.41) and (11.23)), we also compare PPC with the LQR-based consensus algorithm (in abbreviation, LQR) proposed in [1] to demonstrate the superiority of PPC more vividly. As shown in the left panel of Fig. 11.10a, the addition of the predictive mechanism defined in (11.43) yields a drastic increase in convergence speed $D(k)$ towards consensus. All the three methods (classical,

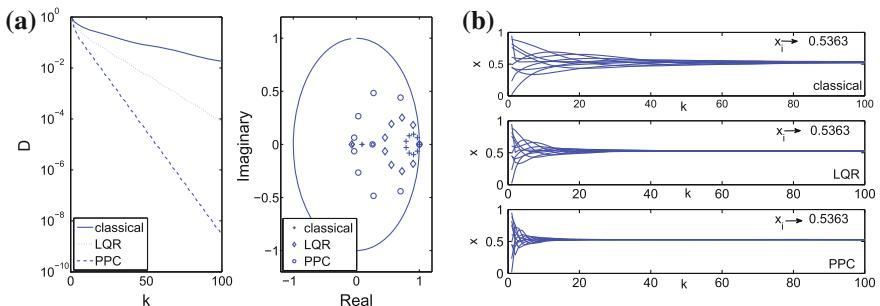


Fig. 11.10 (a) Left consensus index $D_x(k)$; right eigenvalue distribution; (b) Network states trajectory achieving consensus. The parameters of PPC: $r = 1$, $N = 10$, $\varepsilon = 0.1$, $H_u = 3$, $H_p = 5$, $q = 4$; the parameters of LQR [1]: optimal scaling factor $\beta = 0.83$, the weighting matrices $R = I$ and $Q = 3 \cdot I$; the initial states $x_i(0)$, $i = 1, \dots, N$ are selected randomly in $[0, 1]$

Table 11.1 The second largest norm of the eigenvalues for different r

r	1	2	3	4	5
$\rho_1(P_\varepsilon)$	0.9591	0.9527	0.9445	0.9335	0.9187
$\rho_1(P_{LQR})$	0.8793	0.8425	0.7732	0.7424	0.7192
$\rho_1(P_\varepsilon + PPC)$	0.7288	0.6947	0.6502	0.5922	0.5157

LQR and PPC) can achieve the consensus value $\mathbf{1}\mu^T x(0) = \mathbf{1} \cdot 0.5363$ as shown in Fig. 11.10b, where PPC achieves consensus the most quickly. The reason lies in the right panel of Fig. 11.10a, where all the eigenvalues distributions of P_ε , P_{LQR} and $P_\varepsilon + PPC$ are exhibited. Obviously, the spectrum $\rho_1(P_\varepsilon + PPC)$ is smaller than $\rho_1(P_{LQR})$ and $\rho_1(P_\varepsilon)$ (see Definition 11.1 for $\rho_m(\cdot)$). Moreover, consider the networks as shown in Fig. 11.9 with different pinning nodes numbers r , the spectrum $\rho_1(\cdot)$ of the corresponding state matrices of the classical, LQR and PPC laws are demonstrated in Table 11.1. Hereby, Theorem 11.2 is verified, and the superiority of the PPC protocol is thus demonstrated.

It is noted that the advantage of PPC over LQR lies in the fact that PPC makes each node virtually link to the neighbor(s) of its neighbor(s) in future several steps, which increases the virtue connections of the network without adding physical links.

11.2.4 Conclusion

In this section, we proposed a class of pinning predictive controllers (PPCs) for consensus networks to substantially increase their convergence speed towards consensus. The controller does not physically change the network topology or request additional communication channels. Its effectiveness and superiority have been demonstrated through theoretical analysis and numerical simulations.

11.3 Predictive Consensus Control for Double-Integrator Networks with Input Constraints

In this section, we develop a discrete-time consensus protocol for a class of MASs with double-integrator dynamics and switching topologies. Here, the input constraints are considered, and the control horizon is adjustable between one and the prediction horizon H_p , which provides a sufficient degree of freedom for controller design.

11.3.1 Problem Formulation

Consider an MAS with discrete-time double-integrator dynamics and input constraints given as follows [2, 13]:

$$\begin{aligned} q_i(k+1) &= q_i(k) + T p_i(k) + \frac{T^2}{2} u_i(k), \\ p_i(k+1) &= p_i(k) + T u_i(k), \end{aligned} \quad (11.47)$$

$$s.t. \|u_i(k)\| \leq \bar{u}_i, \quad i \in \mathcal{V}, \quad (11.48)$$

where T is the sampling period, $q_i(k), p_i(k) \in \mathbb{R}^m$ denote the position and the velocity of agent i , respectively, $u_i(k) \in \mathbb{R}^m$ is the control input of agent i to be designed at $t = kT$, and $\bar{u}_i > 0$ denotes the upper bound of the Euclidean norm of $u_i(k)$. The dynamics form (11.47) will facilitate the afterwards MPC derivation. A typical MAS problem to be addressed is as follows.

Flocking Problem Develop a decentralized consensus protocol

$$u_i(k) = f_i(p_i(k), p_j(k), q_i(k), q_j(k)), \quad j \in \mathcal{N}_i(k), \quad i \in \mathcal{V}$$

satisfying Eq.(11.47) and

$$\begin{aligned} \lim_{k \rightarrow \infty} \|q_i(k) - q_j(k)\| &= 0, \\ \lim_{k \rightarrow \infty} \|p_i(k) - p_j(k)\| &= 0, \quad \forall i, j \in \mathcal{V}. \end{aligned} \quad (11.49)$$

Note that “flocking” refers to a group of agents moving or migrating together with the same velocity.

11.3.2 Derivation of consensus MPC

Before developing the MPC algorithm, we need a proposition below.

Proposition 11.1 Let $f(U)$ be a quadratic function of

$$U = \text{col}[u_1, \dots, u_n], \quad u_i = [u_{i,1}, \dots, u_{i,m}]^\top \in \mathbb{R}^m. \quad (11.50)$$

Suppose that $f(U)$ can be rewritten as $f(U) = \sum_{l=1}^m f_l(U_l)$, with $U_l := \text{col}[u_{1,l}, \dots, u_{n,l}]$, $f_l(U_l) := \frac{1}{2}(U_l)^\top A_l U_l + \zeta_l U_l + c_l$, $A_l = [a_{i,j,l}] \in \mathbb{R}^{n \times n}$ being a nonsingular symmetric matrix with $a_{i,i,l} > 0$, $\zeta_l := [\zeta_{1,l}, \dots, \zeta_{n,l}]$ being a row vector, c_l being a constant, $l = 1, \dots, m$, $i, j = 1, \dots, n$, and $f(U)$ achieves its minimal value at a unique point $U^* = \text{col}[u_1^*, u_2^*, \dots, u_n^*]$. Then

1. U^* can be calculated by stacking each minimum-value point of $f_l(U_l)$;
2. For the following optimization problem,

$$\min f(U), \quad s.t. \quad \|u_1\| \leq \bar{u}. \quad (11.51)$$

which has a unique optimal point denoted by $\hat{U} = \text{col}[\hat{u}_1, \hat{u}_2, \dots, \hat{u}_n]$, if $A_l \equiv A$ for all $l = 1, \dots, m$, then if $\|u_1^*\| \leq \bar{u}$, $\hat{u}_1 = u_1^*$, otherwise, $\hat{u}_1 = \frac{\bar{u}}{\|u_1^*\|} u_1^*$.

Now, we are ready to develop an MPC scheme to solve **Flocking Problem**. First, we define a quadratic cost function as follows:

$$J_i(k) = J_i^q(k) + J_i^p(k) + J_i^u(k), \quad (11.52)$$

with $J_i^q(k) = \alpha_i \sum_{t=1}^{H_p} \|q_i(k+t|k) - r_i^q(k+t|k)\|^2$, $J_i^p(k) = \beta_i \sum_{t=1}^{H_p} \|p_i(k+t|k) - r_i^p(k+t|k)\|^2$, $J_i^u(k) = \sum_{t=1}^{H_u} \|u_i(k+t-1|k)\|^2$.

Here, integers $H_p, H_u \in \mathbb{N}^+$ denote the prediction and the control horizons, respectively, with $1 \leq H_u \leq H_p$. The signals $r_i^q(k+t|k)$, $r_i^p(k+t|k)$ ($t = 1, 2, \dots, H_p$) denote the position and velocity vector references over the future H_p steps, respectively, positive scalars $\alpha_i, \beta_i \in \mathbb{R}$ are weighting factors and $u_i(k+t-1|k) \in \mathbb{R}^m$ ($t = 1, 2, \dots, H_u$) are the control input sequence over the future H_u steps, where only $u_i(k|k)$ will be the actual control input at step k .

We can obtain the actual control input $u_i(k|k)$ by solving the following quadratic optimization problem:

$$\begin{aligned} & \min J_i(k), \quad i \in \mathcal{V}, \\ & \text{s.t.} \\ & \|u_i(k|k)\| \leq \bar{u}_i. \end{aligned}$$

Denote vectors

$$\begin{aligned} \tilde{q}_l(k) &:= [q_{1,l}(k), q_{2,l}(k), \dots, q_{n,l}(k)]^\top, \\ \tilde{p}_l(k) &:= [p_{1,l}(k), p_{2,l}(k), \dots, p_{n,l}(k)]^\top, \end{aligned} \quad (11.53)$$

with $t = 1, \dots, H_p$, $l \in \mathcal{M} := \{1, 2, \dots, m\}$.

By using $\frac{\partial J_{i,l}(k)}{\partial U_{i,l}(k)} = 0$, one can obtain the optimal MPC input as $U_{i,l}^*(k) = -(P_u^\top Q_i P_u + I_{H_u})^{-1} P_u^\top Q_i [P_x x_{i,l}(k) - r_{i,l}(k)]$. Denoting $\varphi_i = [1, 0, \dots, 0]_{1 \times H_u} \cdot (P_u^\top Q_i P_u + I_{H_u})^{-1} P_u^\top Q_i$, one has

$$u_{i,l}^*(k|k) = -\varphi_i (P_x x_{i,l}(k) - r_{i,l}(k)), \quad (11.54)$$

where the row vector φ_i is independent of $x_{i,l}(k)$. Considering the input constraints (11.48) and Proposition 11.1, one can derive the actual input as

$$u_{i,l}(k|k) = \mathcal{E} u_{i,l}^*(k|k), \quad (11.55)$$

with $\mathcal{E} = \min\{\frac{\bar{u}_i}{\|u_i^*(k|k)\|}, 1\}$.

Lemma 11.2 Given a quadratic function in variables $q_t, p_t, r_t, \hat{r}_0, u_t \in \mathbb{R}$, $t = 0, 1, \dots, H_p$ as follows:

$$J = \alpha \sum_{t=1}^{H_p} (q_t - r_t)^2 + \beta \sum_{t=1}^{H_p} (p_t - \hat{r}_0)^2 + \sum_{t=0}^{H_u-1} u_t^2, \quad (11.56)$$

with $\alpha, \beta \in \mathbb{R}^+$ and $H_u, H_p \in \mathbb{I}^+$ fulfilling $1 \leq H_u \leq H_p$. Assume there exists a positive constant $T \in \mathbb{R}$ such that

1. $q_t = q_{t-1} + T p_{t-1} + \frac{T^2}{2} u_{t-1}, t = 1, \dots, H_p;$
2. $p_t = p_{t-1} + T u_{t-1}, t < H_u;$
3. $p_t = p_{t-1} + T u_{H_u-1}, H_u \leq t \leq H_p;$
4. $r_t = r_{t-1} + T \hat{r}_0, t = 1, \dots, H_p;$
5. J has only one minimum-value point $U = \text{col}[u_0, u_1, \dots, u_{H_u-1}]$,

with constant scalars p_0, q_0, r_0 and $\hat{r}_0 \in \mathbb{R}$. Then, there exist constant scalar series $c_t > 0$ and $\check{c}_t > 0$ such that $u_t = -c_t(q_t - r_t) - \check{c}_t(p_t - \hat{r}_0), t = 0, \dots, H_u - 1$, where c_t, \check{c}_t satisfy

$$c_t T^2 < 2, \check{c}_t T < 2, c_t T < \check{c}_t, \quad (11.57)$$

$$\lim_{T \rightarrow 0} c_t = 0, \lim_{T \rightarrow 0} \check{c}_t = 0, \quad (11.58)$$

$$\lim_{T \rightarrow \infty} c_t = 0, \lim_{T \rightarrow \infty} \check{c}_t = 0. \quad (11.59)$$

Now, the optimization decision value $U = \text{col}[u_0, \dots, u_{H_u-1}]$ of (11.56) is achievable by applying the following iterative Algorithm 1 without calculating the matrix inversion as in Eq.(11.54).

Algorithm 1: Iterative Procedure

- Step (1) Set $h := H_p - H_u + 1$. Compute the parameters $b_{H_u}, c_{H_u-1}, \check{c}_{H_u-1}, \alpha_{H_u-1}, \beta_{H_u-1}, \gamma_{H_u-1}$ (See Appendix in [3]).
 - Step (2) Set the iterative index $i_c = H_u$. Check the iterative index i_c . If $i_c = 1$, go to Step (3); otherwise, set $i_c = i_c - 1$. Calculate the parameters $\sigma = \alpha + \alpha_{i_c-1}, \rho = \beta + \beta_{i_c-1}, b_{i_c}, c_{i_c-1}, \check{c}_{i_c-1}, \alpha_{i_c-1}, \beta_{i_c-1}, \gamma_{i_c-1}$. Return to Step (2);
 - Step (3) Note that $y_0 := q_0 - r_0, z_0 := p_0 - \hat{r}_0$ are constants. If $i_c > H_u$, go to step (4). Otherwise, calculate $u_{i_c-1} = -c_{i_c-1}y_{i_c-1} - \check{c}_{i_c-1}z_{i_c-1}$. Calculating y_{i_c}, z_{i_c} . Set $i_c = i_c + 1$, and return to Step (3).
 - Step (4) Print the minimum point $U = [u_0, \dots, u_{H_u-1}]^T$.
-

11.3.3 Main Result

Now we seek to analytically solve **Flocking Problem**, i.e., the flocking problem. It follows from Eqs.(11.54) and (11.55) that the protocol $u_{i,l}(k|k)$ ($l \in \mathcal{M}$) depends on the reference trajectory vector $r_{i,l}(k)$.

Assumption 10 For each agent i , assume that $p_{j,l}(k+t|k) \equiv p_{j,l}(k)$, $j \in \mathcal{N}_i(k)$, $l \in \mathcal{M}$ and $t = 0, \dots, H_p$. In other words, during the future H_p steps from time instant k , each neighbor j of agent i keeps the same velocity $p_j(k)$.

By Assumption 10, the reference trajectories are

$$\begin{aligned} r_{i,l}^q(k+t|k) &= K_i(\mathcal{G}(k))\tilde{q}_l(k) + t \cdot T \cdot r_{i,l}^p(k|k), \\ r_{i,l}^p(k+t|k) &= K_i(\mathcal{G}(k))\tilde{p}_l(k), \end{aligned} \quad (11.60)$$

where $\tilde{q}_l(k)$, $\tilde{p}_l(k)$ are defined in (11.53), $K_i(\mathcal{G}(k))$ is the i th row of $K(\mathcal{G}(k))$, $l \in \mathcal{M}$, $i \in \mathcal{V}$ and $t = 0, \dots, H_p$.

Hence the protocol (11.54) and (11.55) can be written respectively as

$$u_{i,l}^*(k|k) = -\varphi_i(P_x x_{i,l}(k) - r_{i,l}(k)), \quad (11.61)$$

$$u_{i,l}(k|k) = \Xi u_{i,l}^*(k|k). \quad (11.62)$$

Lemma 11.3 Consider the MAS (11.47) not subject to the constraints (11.48). Under Assumption 10, the control input (11.61) is equivalent to Algorithm 1 in the sense that they both yield the same minimum point

$$u_{i,l}^*(k|k) = -\frac{c_{i_0}}{1+|\mathcal{N}_i(k)|}\mathcal{L}_i(k)\tilde{q}_l(k) - \frac{\check{c}_{i_0}}{1+|\mathcal{N}_i(k)|}\mathcal{L}_i(k)\tilde{p}_l(k), \quad (11.63)$$

where c_{i_0} and \check{c}_{i_0} are positive scalars fulfilling Eqs. (11.57)–(11.59).

Now, we are ready to establish the main result.

Theorem 11.3 Assume that the directed graph topology $\mathcal{G}(k)$ ($k \in \mathbb{I}^+$) of the constrained MAS (11.47) and (11.48) always has a directed spanning tree. If T is selected such that R_k is a nonnegative matrix with positive diagonal entries and $\|S\|_\infty < 1$, then the control law (11.62) solves **Flocking Problem** or Eq. (11.49) if and only if the constraint (11.48) is activated for finitely many times.

11.3.4 Numerical Simulation

Consider an MAS of $n = 5$ agents moving in the two dimensional plane. The proximity network topology switches periodically with period $T = 0.1$ from graph \mathcal{G}_1 (reps. \mathcal{G}_2) to graph \mathcal{G}_2 (resp. \mathcal{G}_1). The graphs \mathcal{G}_1 and \mathcal{G}_2 are defined in Fig. 11.11.

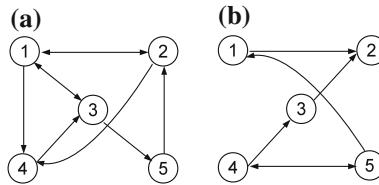


Fig. 11.11 Graph Topologies: **a** \mathcal{G}_1 ; **b** \mathcal{G}_2 . If there exists an arrow from node j to node i , agent i could get access to the information of agent j . Otherwise, the information of agent j is not available to agent i . Note that \mathcal{G}_1 and \mathcal{G}_2 both have a directed spanning tree

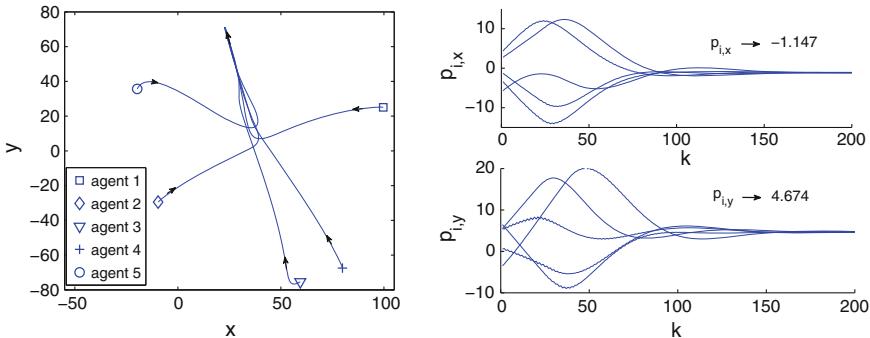


Fig. 11.12 Moving trajectories (positions) and velocity evolution of the MAS

The parameters are $H_p = 10$, $H_u = 8$, $\alpha_i = 1$, $\beta_i = 1$, $\bar{u}_i = 6$, $\forall i \in \mathcal{V}$, and the initial states of all the agents are set as $q_1(0) = [100, 25]$, $p_1(0) = [-3, 1]$; $q_2(0) = [-10, -30]$, $p_2(0) = [4, 5]$; $q_3(0) = [60, -75]$, $p_3(0) = [-6, -4]$; $q_4(0) = [80, -68]$, $p_4(0) = [-1, 5]$; $q_5(0) = [-20, 35]$, $p_5(0) = [2.5, 7]$. The Moving trajectories (positions) and velocities of all the agents are depicted in Fig. 11.12, respectively. It is observed that the velocities achieve flocking while the positions converge to a same manifold in less than 130 steps, and hence **Flocking Problem** is solved. Note that the sampling period $T = 0.1$ is sufficiently small, which fulfills $T \in (0, \bar{T}]$ with \bar{T} given in Theorem 11.3.

11.3.5 Conclusion

An MPC method has been developed to achieve consensus for MASs with double-integrator dynamics, input constraints and switching directed interaction topologies. Under the conditions of the existence of a directed spanning tree, we proved that second-ordered flocking problem is solved by the proposed MPC protocol. More significantly, with the variational control horizon, our method endows sufficient degrees of freedom to accelerate the speed of convergence to consensus, which outperforms the existing methods.

Acknowledgments This work was supported by the National Natural Science Foundation of China (NNSFC) under Grants 61322304, 51328501 and 51120155001, the Natural Science Foundation of Hubei Province under Grant 2012FFA009 and the Research Fund for the Doctoral Program of Higher Education of China under Grant 20130142110050.

References

1. Cao, Y., Ren, W.: LQR-based optimal linear Consensus algorithms. In: Proceedings of American Control Conference, pp. 5204–5209 (2009)
2. Cao, Y., Ren, W.: Sampled-data discrete-time coordination algorithms for double-integrator dynamics under dynamic directed interaction. *Int. J. Control.* **83**, 506–515 (2010)
3. Cheng, Z., Zhang, H.T., Fan, M., Chen, G.: Distributed consensus of multi-agent systems with input constraints: a model predictive control approach. *IEEE Trans. Circuits Syst. I* **62**, 825–834 (2014)
4. Conway, J.H., Guy, R.K.: *The Book of Numbers*. Springer, New York (1996)
5. Couzin, I.D., Krause, J., Franks, N.R., Levin, S.A.: Effective leadership and decision-making in animal groups on the move. *Nature* **433**, 513–516 (2005)
6. Gazi, V., Passino, K.M.: Stability analysis of swarms. *IEEE Trans. Autom. Control* **48**, 692–697 (2003)
7. Horn, R.A., Johnson, C.R.: *Matrix Analysis*. Cambridge University Press, Cambridge (1990)
8. Li, X., Wang, X., Chen, G.: Pinning a complex dynamical network to its equilibrium. *IEEE Trans. Circuits Syst. I* **51**, 297–2087 (2004)
9. Ljung, L.: *System Identification: Theory for the User*. Prentice-Hall Inc., Englewood Cliffs (1999)
10. Maciejowski, J.M.: *Predictive Control with Constraints*. Cambridge University Press, Cambridge (2002)
11. Mayne, D.Q., Rawlings, J.B., Rao, C.V., Scokaer, P.O.M.: Constrained model predictive control: stability and optimality. *Automatica* **36**, 789–814 (2000)
12. Olfati-Saber, R., Murray, R.: Consensus problems in networks of agents with switching topology and time-delays. *IEEE Trans. Autom. Control* **49**, 1520–1533 (2004)
13. Qin, J., Gao, H.: A sufficient condition for convergence of sampled-data consensus for double-integrator dynamics with nonuniform and time-varying communication delays. *IEEE Trans. Autom. Control* **57**, 2417–2422 (2012)
14. Qin, S.J., Badgwell, T.A.: A survey of industrial model predictive control technology. *Control Eng. Pract.* **11**, 733–764 (2003)
15. Ren, W., Beard, R.W.: Consensus seeking in multiagents systems under dynamically changing interaction topologies. *IEEE Trans. Autom. Control* **50**, 655–661 (2005)
16. Ren, W., Beard, R.W., Arkins, E.M.: Information consensus in multivehicle cooperative control. *IEEE Control Syst. Mag.* **71**, 71–82 (2007)
17. Vicsek, T., Czirók, A., Ben-Jacob, E., Cohen, I., Shochet, O.: Novel type of phase transition in a system of self-driven particles. *Phys. Rev. Lett.* **75**, 1226–1229 (1995)
18. Wang, X.F., Li, X., Lu, J.: Control and flocking of networked systems via pinning. *IEEE Circuit Syst. Mag.* **10**, 83–91 (2010)
19. Watts, D.J., Strogatz, S.H.: Collective dynamics of small-world networks. *Nature* **393**, 440–443 (1998)
20. Xiao, L., Boyd, S.: Fast linear iterations for distributed averaging. *Syst. Control Lett.* **53**, 65–78 (2004)

Chapter 12

Convergence, Consensus and Synchronization of Complex Networks via Contraction Theory

Mario di Bernardo, Davide Fiore, Giovanni Russo and Francesco Scafuti

Abstract This chapter reviews several approaches to study convergence of networks of nonlinear dynamical systems based on the use of contraction theory. Rather than studying the properties of the collective asymptotic solution of interest, the strategy focuses on finding sufficient conditions for any pair of trajectories of two agents in the network to converge towards each other. The key tool is the study, in an appropriate metric, of the matrix measure of the agents' or network Jacobian. The effectiveness of the proposed approach is illustrated via a set of representative examples.

12.1 Introduction

The problem of steering the collective behaviour of a network of dynamical agents towards a desired common target solution is a fundamental problem in network science and control theory [1–3]. A typical problem is that of achieving consensus or synchronization in a network of linear or nonlinear systems. Here, the key challenge, once the coupling function among the nodes has been selected, is to prove convergence of all nodes towards the desired common asymptotic behaviour. This convergence problem is usually solved locally by means of the Master Stability Function Approach (MSF) [4] or globally via Lyapunov stability theory [5].

Global convergence is extremely useful particularly when the agents are subject to high level of noise or perturbations as it is often the case in applications; think for example of the recent application of synchronization to biological systems [6].

M. di Bernardo (✉) · D. Fiore · G. Russo · F. Scafuti
Department of Electrical Engineering and Information Technology,
University of Naples Federico II, Via Claudio 21, 80125 Naples, Italy
e-mail: mario.dibernardo@unina.it

D. Fiore
e-mail: davide.fiore@unina.it

G. Russo
e-mail: giovanni.russo2@unina.it

F. Scafuti
e-mail: francesco.scafuti@unina.it

In this case, it is essential to possess alternative strategies to study convergence of all nodes towards the desired synchronous state.

Contraction theory has been recently proposed as an effective approach to study convergence between trajectories of a dynamical systems (see for example [7] and the more recent papers [8–13]). The idea is to characterize within some metric the distance between trajectories and to prove that the matrix measure of the system Jacobian is uniformly negative in that metric over some connected forward-invariant set of the state space. Indeed, it can then be proved that this condition implies global exponential incremental stability over the set of interest. Note that, using contraction, it is possible to use different measures to study the properties of the Jacobian including non-Euclidean ones (e.g., μ_1 , μ_∞ etc.). Therefore, this approach does not require finding explicitly quadratic Lyapunov functions for the system of interest.

It has been shown that contraction is an extremely useful property to analyze convergence in networks and study problems such as the emergence of synchronization or consensus [7, 8, 14–17]. Indeed, all trajectories of a contracting system can be shown to exponentially converge towards each other asymptotically. Therefore as shown in [14], this property can be effectively exploited to give conditions for the synchronization of a network of dynamical systems of interest. It has been noted that non Euclidean matrix measures can be useful to construct an algorithmic approach to prove contraction [8] and to prove efficiently convergence in biological networks [9, 18]. Also, a hierarchical approach to study convergence using contraction was discussed in [19].

For an historical overview of results closely related to contraction see [7, 20–26]. See also [27] for a more exhaustive list of related references.

The aim of this paper is to expound a self-contained overview of contraction and its application to synchronization and consensus problems. As the relevant literature is growing fast, particular attention will be given to the approaches based on contraction developed in previous work by some of the authors.

12.2 Mathematical Preliminaries

Throughout this chapter, $\mathbb{R}^{n \times n}$ denotes the set of all $n \times n$ real matrices. We now introduce the matrix measure associated to a matrix $A \in \mathbb{R}^{n \times n}$, that is the function $\mu_i(\cdot) : \mathbb{R}^{n \times n} \rightarrow \mathbb{R}$ defined in the following way.

Definition 12.1 (*Matrix measure* [28]) Given a real matrix $A \in \mathbb{R}^{n \times n}$ the corresponding *matrix measure* $\mu(A)$ is defined as

$$\mu_i(A) = \lim_{h \rightarrow 0^+} \frac{\|I + hA\|_i - 1}{h}$$

The measure of a matrix A can be thought of as the directional derivative of the induced matrix norm function $\|\cdot\|_i$, evaluated at the identity matrix I , in the direction

Table 12.1 Vector norms, induced matrix norms and matrix measure

Vector norm	Induced matrix norm	Matrix measure
$ x _1 = \sum_{i=1}^n x_i $	$\ A\ _1 = \max_j \sum_{i=1}^n a_{ij} $	$\mu_1(A) = \max_j [a_{jj} + \sum_{i \neq j} a_{ij}]$
$ x _2 = (\sum_{i=1}^n x_i ^2)^{1/2}$	$\ A\ _2 = [\lambda_{\max}(A^T A)]^{1/2}$	$\mu_2(A) = \lambda_{\max} \frac{A^T + A}{2}$
$ x _\infty = \max_i x_i $	$\ A\ _\infty = \max_i \sum_{j=1}^n a_{ij} $	$\mu_\infty(A) = \max_i [a_{ii} + \sum_{j \neq i} a_{ij}]$

of A . In Table 12.1, three different types of vector norm are considered together with the respective induced matrix norm and matrix measure. In general, given a constant invertible matrix P , the matrix measure $\mu_{P,i}$ induced by the weighted vector norm $|x|_P = |Px|$ can also be defined as $\mu_{P,i} = \mu_i(PAP^{-1})$.

Definition 12.2 (\mathcal{K} function [29]) A scalar continuous function $\alpha(r)$ defined for $r \in [0, a)$ is said to belong to class \mathcal{K} if it is strictly increasing and $\alpha(0) = 0$. It is said to belong to class \mathcal{K}_∞ if it is defined for all $r \geq 0$ and

$$\lim_{r \rightarrow \infty} \alpha(r) = \infty$$

Definition 12.3 (\mathcal{KL} function [29]) A scalar continuous function $\beta(r, t)$ defined for $r \in [0, a)$ and $t \in [0, \infty)$ is said to belong to class \mathcal{KL} if, for each fixed \bar{t} , $\beta(r, \bar{t})$ belongs to class \mathcal{K} with respect to r and for each fixed \bar{r} , the mapping $\beta(\bar{r}, t)$ is decreasing with respect to t and

$$\lim_{t \rightarrow \infty} \beta(\bar{r}, t) = 0$$

12.3 Contraction Theory and Incremental Stability

Next, we provide some definitions concerning the stability of a nonlinear dynamical system of the form:

$$\dot{x} = f(x, t) \tag{12.1}$$

where $x \in \mathbb{R}^n$, $t \in \mathbb{R}$ and $f : \mathbb{R}^n \times \mathbb{R} \mapsto \mathbb{R}^n$. We denote with $\phi(t - t_0, t_0, x_0)$ the value of the solution $x(t)$ evaluated at time t of the differential equation (12.1) with initial value $x(t_0) = x_0$. We say that a set $\mathbb{C} \subseteq \mathbb{R}^n$ is a *forward invariant* set for system (12.1), if for every $t_0 \geq 0$, $x(t_0) = x_0 \in \mathbb{C}$ implies $\phi(t - t_0, t_0, x_0) \in \mathbb{C}$ for all $t \geq t_0$.

Specifically, we are interested in characterizing the stability of any two arbitrary solutions of the system with respect to one another. We refer to this property as *incremental stability*, using the definition first presented in [23].

Definition 12.4 (*Incremental stability* [23]) System (12.1) is said to be *Incrementally Asymptotically Stable* (δ AS) in a forward invariant set $\mathbb{C} \subseteq \mathbb{R}^n$, if there exists a class \mathcal{KL} function β , such that, for any $x_0, z_0 \in \mathbb{C}$ and $t \geq t_0$, any two of its trajectories, say $x(t) = \phi(t - t_0, t_0, x_0)$ and $z(t) = \phi(t - t_0, t_0, z_0)$, verify

$$|x(t) - z(t)| \leq \beta(|x_0 - z_0|, t - t_0)$$

Moreover, if there exist real numbers $c > 0$, $K \geq 1$ such that for all $t \geq 0$

$$|x(t) - z(t)| \leq K |x_0 - z_0| e^{-c(t-t_0)}$$

we say that system (12.1) is *Incrementally Exponential Stable* (δ ES). Finally, if $\mathbb{C} \equiv \mathbb{R}^n$, then system (12.1) is said to be *Globally Incrementally Asymptotically Stable* (δ GAS) or *Globally Incrementally Exponentially Stable* (δ GES), respectively.

An effective approach to prove incremental stability is to use the concept of contracting system as expounded in [7].

Definition 12.5 (*Infinitesimal Contraction*) System (12.1) is said to be infinitesimally contracting on a set $\mathcal{C} \subseteq \mathbb{R}^n$ if there exists a norm in \mathcal{C} , with associated matrix measure $\mu_i(\cdot)$, such that, for some constant $c > 0$ (termed as contraction rate), it holds that:

$$\mu(J(x, t)) \leq -c, \quad \forall x \in \mathcal{C}, \quad \forall t \geq 0.$$

In [7, 9] it is shown that the following sufficient condition for δ ES holds.

Theorem 12.1 ([9]) Suppose \mathcal{C} is a convex subset of \mathbb{R}^n and that system (12.1) is infinitesimally contracting with contraction rate c . Then for every two solutions $x(t) = \varphi(t, 0, x_0)$, $z(t) = \varphi(t, 0, z_0)$ it holds that:

$$|x(t) - z(t)| \leq e^{-ct} |x_0 - z_0| \tag{12.2}$$

Proof See [9]. □

Remark 12.1 From Theorem 12.1, it is clear that if $\mathcal{C} \equiv \mathbb{R}^n$, then system (12.1) is δ GES.

12.3.1 Example

As a representative example, we consider the model

$$\begin{cases} \dot{x}_1 = x_1 (x_1^2 + x_2^2 - 1) + x_2 \\ \dot{x}_2 = -x_1 + x_2 (x_1^2 + x_2^2 - 1) \end{cases}$$

The Jacobian matrix of the system is

$$J = \begin{bmatrix} -1 + 3x_1^2 + x_2^2 & 1 + 2x_1x_2 \\ -1 + 2x_1x_2 & -1 + x_1^2 + 3x_2^2 \end{bmatrix}$$

and its symmetric part is given by

$$J_s := \frac{J + J^T}{2} = \begin{bmatrix} -1 + 3x_1^2 + x_2^2 & 2x_1x_2 \\ 2x_1x_2 & -1 + x_1^2 + 3x_2^2 \end{bmatrix}$$

Now, computing the eigenvalues $\lambda_i(J_s)$ yields

$$\begin{aligned}\lambda_1 &= x_1^2 + x_2^2 - 1 \\ \lambda_2 &= 3x_1^2 + 3x_2^2 - 1\end{aligned}$$

It is simple to verify that

$$\lambda_2 > \lambda_1 \quad \forall x \in \mathbb{R}^2$$

Thus $\mu_2(J_s) < 0$ implies

$$3x_1^2 + 3x_2^2 - 1 < 0$$

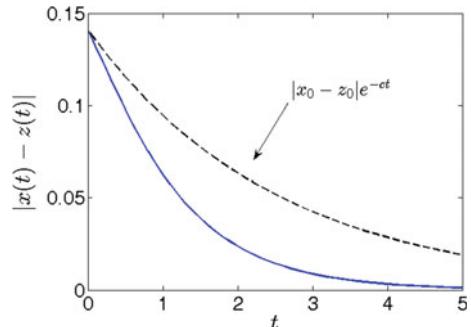
i.e. $\mu_2(J_s) < 0$ for all x in the circle of radius $\sqrt{3}/3$ centred at the origin. Thus, we select the set \mathcal{C} as

$$\mathcal{C} := \{x \in \mathbb{R}^2 : |x| < \sqrt{3}/3\}$$

Since the origin is an equilibrium point and it belongs to \mathcal{C} , we can conclude that the set \mathcal{C} is a forward invariant set for the system.

In Fig. 12.1 the norm of the error is showed, between two trajectories $x(t)$ and $z(t)$ rooted in different initial conditions $x_0 = [0.4 \ 0.2]^T$, $z_0 = [0.3 \ 0.1]^T$. For the sake of comparison, the behaviour of $|x_0 - z_0|e^{-ct}$ is also depicted (dashed black line with $c = 0.4$)

Fig. 12.1 Norm of the error (blue solid line) between two trajectories $x(t)$ and $z(t)$ rooted in different initial conditions $x_0 = [0.4 \ 0.2]^T$, $z_0 = [0.3 \ 0.1]^T$. For the sake of comparison, the behaviour of $|x_0 - z_0|e^{-ct}$ is also depicted (dashed black line with $c = 0.4$)



The contraction rate c can be estimated as

$$c = \min\{|3x_{0,1}^2 + 3x_{0,2}^2 - 1|, |3z_{0,1}^2 + 3z_{0,2}^2 - 1|\}$$

where $x_{0,i}$ and $z_{0,i}$ are the i th component of the vectors x_0 and z_0 , respectively. Thus, we have $c = 0.4$.

12.3.2 Properties of Contracting Systems

Contracting systems have been shown to possess several useful properties. Consider the cascade system of the following form:

$$\begin{aligned}\dot{x} &= f(x, t) \\ \dot{y} &= g(x, y, t)\end{aligned}$$

where $x(t) \in \mathcal{C}_1 \subseteq \mathbb{R}^{n_1}$ and $y(t) \in \mathcal{C}_2 \subseteq \mathbb{R}^{n_2}$ for all t .

The Jacobian of this system is

$$J = \begin{bmatrix} A & 0 \\ B & C \end{bmatrix} \quad (12.3)$$

where we have written the Jacobian of f with respect to x as $A(x, t) = \frac{\partial f}{\partial x}(x, t)$, the Jacobian of g with respect to x as $B(x, y, t) = \frac{\partial g}{\partial x}(x, y, t)$, and the Jacobian of g with respect to y as $C(x, y, t) = \frac{\partial g}{\partial y}(x, y, t)$.

Theorem 12.2 ([9]) Suppose that

- the system $\dot{x} = f(x, t)$ is infinitesimally contracting with contraction rate c_1
- the system $\dot{y} = g(x, y, t)$ is infinitesimally contracting with contraction rate c_2 when x is viewed as a parameter
- the mixed Jacobian $B(x, y, t)$ is bounded, that is $\|B(x, y, t)\| \leq k$, $k > 0$

then the cascade system is infinitesimally contracting. More precisely, pick any two positive numbers p_1 and p_2 such that $c_1 - \frac{p_2}{p_1}k > 0$ and let $c := \min \left\{ c_1 - \frac{p_2}{p_1}k, c_2 \right\}$ then $\mu(J) \leq -c$.

Proof See [9]. □

Another useful property, often exploited in applications of contraction theory to synchronization and entrainment problems, refers to the case where a contracting system is forced by an external periodic signal. In particular, given a number $T > 0$, we will say that system (12.1) is *T-periodic* if it holds that

$$f(x, t + T) = f(x, t) \quad \forall t \geq 0$$

Notice that a system $\dot{x} = f(x, u(t))$ with input $u(t)$ is T -periodic if $u(t)$ is itself a periodic function of period T . We can then state the following basic result about existence and stability of periodic orbits.

Theorem 12.3 ([9]) Suppose that

- \mathcal{C} is a closed convex subset of \mathbb{R}^n ;
- f is infinitisimally contracting with contraction rate c ;
- f is T -periodic.

Then there is a unique periodic orbit $\widehat{\omega}$ in \mathcal{C} of (12.1) of period T and, for every solution $x(t)$ starting in \mathcal{C} , it holds that $\text{dist}(x(t), \widehat{\omega}) \rightarrow 0$ as $t \rightarrow \infty$.

This property was used in [9] to prove global entrainment of transcriptional biological networks and can be effectively used whenever the goal is to prove entrainability of a system or network of interest.

12.4 Synchronization and Contraction Theory

Next we discuss how contraction analysis can be used to prove convergence in networks of nonlinear systems and prove their synchronization. We consider a generic homogeneous network whose nodes' dynamics can be described by the following set of differential equations

$$\dot{x}_i = f(x_i, t) + \tilde{h}_i(x_1, \dots, x_N, t), \quad i = 1, \dots, N \quad (12.4)$$

where $x_i \in \mathbb{R}^n$ is the state vector of node i , the vector field $f : \mathbb{R}^n \times \mathbb{R}^+ \rightarrow \mathbb{R}^n$ describes the intrinsic dynamics of the i th node, and the function $\tilde{h}_i : \mathbb{R}^n \times \dots \times \mathbb{R}^n \times \mathbb{R}^+ \rightarrow \mathbb{R}^n$ represents the coupling function describing how the i th node interacts with the other nodes in the network. We assume the network structure is connected.

In the case of diffusive coupling, we can write

$$\dot{x}_i = f(x_i, t) + \sum_{j \in \mathcal{N}_i} [h(x_j) - h(x_i)] \quad i = 1, \dots, N$$

where $h : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is some nonlinear coupling function, and \mathcal{N}_i is the set of the neighbours of node i , that is the set of nodes connected to node i . If the coupling is linear and diffusive, the network equations become

$$\begin{aligned} \dot{x}_i &= f(x_i, t) + \sigma \sum_{j \in \mathcal{N}_i} (x_j - x_i) \\ &= f(x_i, t) - \sigma \sum_{j=1}^N l_{ij} x_j \end{aligned} \quad (12.5)$$

for $i = 1, \dots, N$, where l_{ij} is the ij th element of the Laplacian matrix L associated with the graph describing the network topology, and σ is the overall coupling strength.

In some cases it is useful to rewrite the network equations in block form

$$\dot{X} = F(X) - \sigma(L \otimes I_n)X \quad (12.6)$$

where $X = [x_1^T, \dots, x_n^T]^T \in \mathbb{R}^{nN}$, $F(X) = [f^T(x_1), \dots, f^T(x_N)]^T$ is the stack vector of all node vector fields, and \otimes denotes the Kronecker product.

Definition 12.6 A network is said to achieve (asymptotic) *synchronization* if and only if

$$\lim_{t \rightarrow +\infty} (x_i(t) - x_j(t)) = 0 \quad \forall i, j$$

That is all network trajectories converge towards the synchronization manifold

$$\mathcal{S} := \left\{ \begin{bmatrix} x_1^T, \dots, x_N^T \end{bmatrix} \in \mathbb{R}^{nN} : x_1 = \dots = x_N \right\} \quad (12.7)$$

Furthermore a network is said to achieve *consensus* when $x_1 = \dots = x_N$, and $\dot{x}_i = 0$ for all i (i.e. all node converge onto some constant steady-state value). For further details about synchronization and consensus see [30, 31] and [32, 33].

Several alternative approaches are available to prove synchronization of a complex network such as (12.6) using contraction theory, each with its own advantages and limitations. In what follows we will expound some of the most common methodologies presented in the literature highlighting their advantages and limitations. Specifically we will discuss the following strategies: (i) virtual system method; (ii) contraction to flow-invariant subspaces; (iii) hierarchical approach.

12.5 Virtual System Method

The virtual system method, firstly presented in [7, 14], is based on proving synchronization by constructing a virtual (or *auxiliary*) system whose particular solutions are the solutions of each of the nodes in the network. Specifically, the virtual system has all the solutions $x_i(t)$ of the original network as its own solutions, and a particular solution, say $y_\infty(t)$, that verifies a smooth specific property. Then, if the virtual system is proved to be contracting then all its trajectories converge exponentially towards each other and towards y_∞ . Therefore, all trajectories of the nodes in the network verify the same property exponentially. If a virtual system exists, then the original system (or network) is said to be *partially contracting*.

To illustrate the previous idea let us consider, as a simple example, a network of two diffusively coupled identical nodes

$$\begin{cases} \dot{x}_1 = f(x_1) + h(x_2) - h(x_1) \\ \dot{x}_2 = f(x_2) + h(x_1) - h(x_2) \end{cases}$$

To prove convergence of x_1 and x_2 towards each other, we can construct the following virtual system

$$\dot{y} = f(y) - 2h(y) + h(x_1) + h(x_2)$$

Indeed, it is immediate to verify that this system has $x_1(t)$ and $x_2(t)$ as particular solutions, and if it is contracting, that is if for all x_1 and x_2 ,

$$\mu \left(\frac{\partial f(y)}{\partial y} - 2 \frac{\partial h(y)}{\partial y} \right) \leq -c \quad \forall y \in \mathcal{C}, \quad \forall t \geq 0$$

then the two node trajectories exponentially converge towards each other in \mathcal{C} and the network synchronizes.

12.5.1 Synchronization of Networks with an All-to-All Topology

As discussed in [14], constructing virtual systems is not, in general, an easy task given a generic network structure. It becomes immediate in the case of fully connected networks. Specifically, consider the all-to-all network

$$\begin{aligned} \dot{x}_i &= f(x_i) + \sum_{j=1}^N [h(x_j) - h(x_i)] \\ &= f(x_i) - Nh(x_i) + \sum_{j=1}^N h(x_j) \end{aligned}$$

for $i = 1, \dots, N$. The virtual system can then be selected as

$$\dot{y} = f(y) - Nh(y) + \sum_{j=1}^N h(x_j)$$

and it is contracting if some matrix measure μ exists such that

$$\mu \left(\frac{\partial f(y)}{\partial y} - N \frac{\partial h(y)}{\partial y} \right) \leq -c \quad \forall y \in \mathcal{C}, \quad \forall t \geq 0$$

Unfortunately the simplicity of the method is lost when more generic topologies are considered.

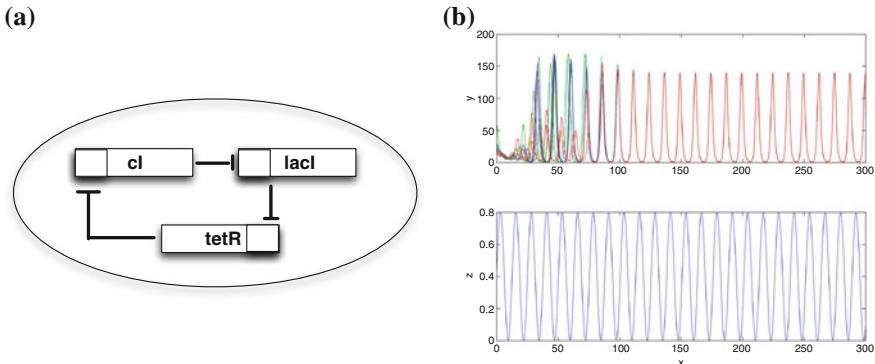


Fig. 12.2 A schematic representation of the three-genes Repressilator circuit (a). Simulation results showing that the population of Repressilators synchronize to a common steady-state solution (Top panel) having the same period as $u(t) = 0.4 + 0.4 \sin(0.5t)$ (Bottom panel) (b) (Reproduced from Figs. 8 and 13 in [9], with permission.)

12.5.2 Example

As a representative example of application of the virtual system approach let us consider the all-to-all network of biological oscillators presented in [9].

The Repressilator is a synthetic biological circuit of three genes inhibiting each other in a cyclic way [34]. As shown in Fig. 12.2a, gene *lacI* (associated to the state variable c_i in our model) expresses protein *LacI* (C_i), which inhibits transcription of gene *tetR* (a_i). This translates into protein *TetR* (A_i), which inhibits transcription of gene *cl* (b_i). Finally, the protein *Cl* (B_i) translated from *cl* inhibits expression of *lacI*, completing the cycle. The resulting mathematical model for the network is

$$\begin{aligned}
 \dot{a}_i &= -a_i + \frac{\alpha}{1 + C_i^2} \\
 \dot{b}_i &= -b_i + \frac{\alpha}{1 + A_i^2} \\
 \dot{c}_i &= -c_i + \frac{\alpha}{1 + B_i^2} + \frac{kS_i}{1 + S_i} \\
 \dot{A}_i &= \beta_A a_i - d_A A_i \\
 \dot{B}_i &= \beta_B b_i - d_B B_i \\
 \dot{C}_i &= \beta_C c_i - d_C C_i \\
 \dot{S}_i &= -k_{s0} S_i + k_{s1} A_i - \eta(S_i - S_e) \\
 \dot{S}_e &= -k_{se} S_e + \eta_{ext} \sum_{j=1}^N (S_j - S_e) + u(t)
 \end{aligned} \tag{12.8}$$

For further details on the mathematical model see [9, 34, 35].

The network is an all-to-all network of identical nodes, hence the virtual system can be chosen as having the same dynamics as the individual Repressilator circuit, forced by the external coupling signal S_e , i.e.

$$\begin{aligned}\dot{a} &= -a + \frac{\alpha}{1+C^2} \\ \dot{b} &= -b + \frac{\alpha}{1+A^2} \\ \dot{c} &= -c + \frac{\alpha}{1+B^2} + \frac{kS}{1+S} \\ \dot{A} &= \beta_A a - d_A A \\ \dot{B} &= \beta_B b - d_B B \\ \dot{C} &= \beta_C c - d_C C \\ \dot{S} &= -k_{s0} S + k_{s1} A - \eta(S - S_e) \\ \dot{S}_e &= -k_{se} S_e + \eta_{ext}(S_1 + \dots + S_N) - \eta_{ext} N S_e + u(t)\end{aligned}\tag{12.9}$$

Indeed, by direct inspection it is easy to check that, by substituting the state variables of the nodes dynamics for the virtual variables (i.e., $[a_i, b_i, c_i, A_i, B_i, C_i, S_i, S_e]$ for $[a, b, c, A, B, C, S, S_e]$), the equations of each Repressilator circuit in the network can be obtained. In this sense, the virtual system embeds the trajectories of all network oscillators as particular solutions. Thus, contraction of the virtual system (12.9) implies synchronization of (12.8). Differentiation of (12.9) yields the Jacobian matrix

$$J = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & f_1(C) & 0 & 0 \\ 0 & -1 & 0 & f_1(A) & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & f_1(B) & 0 & f_2(S) & 0 \\ \beta_A & 0 & 0 & -d_A & 0 & 0 & 0 & 0 \\ 0 & \beta_B & 0 & 0 & -d_B & 0 & 0 & 0 \\ 0 & 0 & \beta_C & 0 & 0 & -d_C & 0 & 0 \\ 0 & 0 & 0 & k_{s1} & 0 & 0 & -k_{s0} - \eta & \eta \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -k_q \end{bmatrix}$$

where f_1 and f_2 denote the partial derivatives of decreasing and increasing Hill functions with respect to the state variable of interest and $k_q = k_{se} + \eta_{ext} N$.

Note that the Jacobian matrix J has the (transposed) structure of the cascade system given in (12.3), i.e.

$$J = \begin{bmatrix} A & B \\ 0 & C \end{bmatrix}$$

with

$$A = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & f_1(C) & 0 \\ 0 & -1 & 0 & f_1(A) & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & f_1(B) & 0 & f_2(S) \\ \beta_A & 0 & 0 & -d_A & 0 & 0 & 0 \\ 0 & \beta_B & 0 & 0 & -d_B & 0 & 0 \\ 0 & 0 & \beta_C & 0 & 0 & -d_C & 0 \\ 0 & 0 & 0 & k_{s1} & 0 & 0 & -k_{s0} - \eta \end{bmatrix},$$

$B = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ \eta]^T$, $C = -k_q$. Theorem 12.2 can be used to prove contraction of the virtual system. Specifically, it suffices to prove that there exist two matrix measures, μ_* and μ_{**} such that

1. $\mu_*(A) \leq -c_1$, with $c_1 > 0$
2. $\mu_{**}(C) \leq -c_2$, with $c_2 > 0$
3. $\|B\| \leq k$, with $k > 0$

Clearly, since k_q and η are positive real parameters, the second and third conditions above are satisfied. On the other hand, the matrix measure of A can be made negative by choosing adequate values of the network parameters (see [9] for further details). Thus, we can conclude that (12.9) is contracting, and in turn this implies that all the nodes of (12.8) synchronize on the same solution, as can be noted in Fig. 12.2b. Furthermore, if $u(t)$ is a T -periodic function, the N interconnected dynamical systems entrain onto a T -periodic trajectory (see Theorem 12.3).

12.5.3 Synchronization of Networks with a Generic Topology

Constructing the virtual system for more generic network structures is cumbersome. A possible approach is presented in [14] where it is noted that in some cases the virtual system for a generic network structure can be written as

$$\dot{Y} = F(Y) - \sigma(L \otimes I_n)Y - (1_{N \times N} \otimes K_0)(Y - X) \quad (12.10)$$

where $Y = [y_1^T, \dots, y_n^T]^T \in \mathbb{R}^{nN}$ is the set of virtual state variables, K_0 is some constant symmetric positive definite matrix and $1_{N \times N}$ is the $N \times N$ matrix whose elements are all equal to 1.

It can again be noticed that this virtual system is such that it embeds all the solutions of the original network as particular solutions, in fact for $Y = X$ (12.10) gives (12.6). Furthermore this system admits the particular solution $y_1 = \dots = y_N = y_\infty$ with

$$\dot{y}_\infty = f(y_\infty) - nK_0y_\infty + K_0 \sum_{j=1}^N x_j(t)$$

Therefore if the virtual system is proved to be contracting then all its trajectories converge exponentially towards each other and towards y_∞ , this in turns implies synchronization of the network (12.6).

Notice that K_0 is a virtual quantity used to prove contraction of the virtual system, and thus it cannot affect the synchronization of the original network.

For a full proof of the virtual system method see [14], while the link between this approach and the Master Stability Function [4] is discussed in [36]. The most notable difference is that, while the MSF approach guarantees *local* transversal stability of the synchronization manifold, contraction analysis gives a *global* result in the region of interest.

12.6 Convergence to a Flow-Invariant Linear Subspace

Contraction theory can also be used to guarantee convergence of system trajectories towards some flow-invariant linear manifold [15]. Hence, synchronization can be proved by using contraction analysis to prove global convergence of the network nodes towards the synchronization manifold defined in (12.7). A more general description of this approach is presented in [15], where the problem of concurrent synchronization is discussed.

Assume that for the dynamical system $\dot{x} = f(x, t)$ there exists a linear flow-invariant subspace \mathcal{M} of dimension p . Let $[e_1, \dots, e_n]$ be an orthonormal basis of \mathbb{R}^n where the first p vectors form a basis of \mathcal{M} and the last $n - p$ a basis of \mathcal{M}^\perp . Define now the following matrices

$$U := \begin{bmatrix} e_1^T \\ \vdots \\ e_p^T \end{bmatrix}, \quad V := \begin{bmatrix} e_{p+1}^T \\ \vdots \\ e_n^T \end{bmatrix}$$

The following result holds.

Theorem 12.4 All trajectories of $\dot{x} = f(x, t)$ globally exponentially converge towards \mathcal{M} if there exist some matrix measure and some $c > 0$ such that

$$\mu(V J(x, t) V^T) \leq -c \quad \forall x \in \mathbb{R}^n, \quad \forall t \geq 0$$

where $J(x, t)$ is the system Jacobian.

A simple proof of the previous theorem can be found in [37] as a generalization of the one presented in [15].

The previous theorem can be used to prove synchronization of a complex network by considering as linear subspace \mathcal{M} the synchronization manifold \mathcal{S} in (12.7) and

V as the orthonormal matrix spanning \mathcal{S}^\perp . The matrix V can be obtained as the orthonormalization of the following $n(N - 1) \times nN$ matrix

$$\begin{bmatrix} I_n & -I_n & O_n & O_n & \dots & O_n \\ O_n & I_n & -I_n & O_n & \dots & O_n \\ \vdots & & \ddots & \ddots & & \vdots \\ O_n & O_n & \dots & I_n & -I_n & O_n \\ O_n & O_n & \dots & O_n & I_n & -I_n \end{bmatrix}$$

where I_n and O_n are respectively $n \times n$ identity and zero matrices.

Let $F(X, t)$ be the stack of all the node vector fields and $H(X, t)$ the stack vector of all the coupling functions \tilde{h}_i in (12.4).

Using the concept of contraction towards the synchronization subspace, we can prove the following result [37].

Theorem 12.5 *Assume that for network (12.4) the subspace \mathcal{S} exists, then network synchronization is attained if there exists some matrix measure such that*

$$\mu\left(V \frac{\partial F}{\partial X} V^T\right) < -\mu\left(V \frac{\partial H}{\partial X} V^T\right) \quad \forall x \in \mathbb{R}^n, \quad \forall t \geq 0$$

Proof Indeed, from Theorem 12.4 we have that all network trajectories converge towards the synchronization subspace \mathcal{S} if

$$\mu\left(V \left(\frac{\partial F}{\partial X} + \frac{\partial H}{\partial X}\right) V^T\right)$$

is uniformly negative definite. Now, from the the subadditivity property of matrix measure [28] we have that the above condition is satisfied if

$$\mu\left(V \frac{\partial F}{\partial X} V^T\right) + \mu\left(V \frac{\partial H}{\partial X} V^T\right)$$

is uniformly negative definite. This proves the result. \square

If we consider a network of diffusively coupled nonlinear nodes as in (12.5)–(12.6), the following result holds from applying contraction using Euclidean matrix measures. (An historical overview of similar results and further proofs can be found in [12, 13]. Here we report alternative proofs based on contraction to make the chapter self contained.)

Theorem 12.6 *All network trajectories converge towards \mathcal{S} if*

$$\max_{x,t} \lambda_{\max}\left(\frac{\partial f}{\partial x}\right) \leq \sigma \lambda_2 \quad \forall x \in \mathbb{R}^n, \quad \forall t \geq 0$$

where λ_2 is the algebraic connectivity of the network.

Proof Let V be the orthonormal matrix spanning \mathcal{S}^\perp , this can be chosen as

$$V = (Q \otimes I_n)^T, \quad (12.11)$$

where Q is the orthonormal $N - 1 \times N - 1$ matrix such that

$$Q^T L Q = \Lambda$$

and Λ is the $N - 1 \times N - 1$ diagonal matrix containing on its main diagonal the eigenvalues $\lambda_2, \dots, \lambda_N$ of L .

Now, let J be the Jacobian of (12.6) then from Theorem 12.4 we have that all network trajectories converge towards \mathcal{S} if $\mu_2(V J V^T)$ is uniformly negative definite. That is if

$$\mu_2\left(V \frac{\partial F}{\partial X} V^T - V(\sigma(L \otimes I_n))V^T\right) \leq -c, \quad c > 0$$

Then

$$\mu_2\left(V \frac{\partial F}{\partial X} V^T - V(\sigma(L \otimes I_n))V^T\right) \leq \mu_2\left(V \frac{\partial F}{\partial X} V^T\right) + \mu_2\left(-V(\sigma(L \otimes I_n))V^T\right) \quad (12.12)$$

Remembering that for the Kronecker product the following property holds

$$(A \otimes B)(C \otimes D) = (AC) \otimes (BD),$$

we have

$$\begin{aligned} V(L \otimes I_n)V^T &= (Q \otimes I_n)^T(L \otimes I_n)(Q \otimes I_n) = \\ &= (Q^T \otimes I_n)(L \otimes I_n)(Q \otimes I_n) = \\ &= (Q^T L \otimes I_n)(Q \otimes I_n) = \\ &= Q^T L Q \otimes I_n = \\ &= \Lambda \otimes I_n \end{aligned}$$

Therefore

$$\mu_2\left(-V(\sigma(L \otimes I_n))V^T\right) = \sigma \mu_2(-\Lambda) = -\sigma \lambda_2$$

Furthermore

$$\mu_2\left(V \frac{\partial F}{\partial X} V^T\right) = \lambda_{\max}\left(V \left[\frac{\partial F}{\partial X}\right]_s V^T\right)$$

where $\left[\frac{\partial F}{\partial X} \right]_s$ denotes the symmetric part of $\frac{\partial F}{\partial X}$. To evaluate the above matrix measure, consider the quadratic form

$$v^T V \left[\frac{\partial F}{\partial X} \right]_s V^T v = a^T \left[\frac{\partial F}{\partial X} \right]_s a.$$

Notice that, since the matrix $\frac{\partial F}{\partial X}$ is a block diagonal matrix, we have for any $a \neq 0$

$$\min_{x,t} \lambda_{\min} \left(\frac{\partial f}{\partial x} \right) a^T a \leq a^T \frac{\partial F}{\partial X} a \leq \max_{x,t} \lambda_{\max} \left(\frac{\partial f}{\partial x} \right) a^T a,$$

where $\frac{\partial f}{\partial x}$ is the Jacobian of the intrinsic dynamics of the node. On the other hand $a^T a = v^T V V^T v = v^T v$. Thus (12.12) become

$$\mu_2 \left(V \frac{\partial F}{\partial X} V^T - V(\sigma(L \otimes I_n))V^T \right) \leq \max_{x,t} \lambda_{\max} \left(\frac{\partial f}{\partial x} \right) - \sigma \lambda_2.$$

Since the above quantity is uniformly negative definite from the hypotheses, we have that all network trajectories globally exponentially converge towards \mathcal{S} . This proves the result. \square

In the specific case where all the network nodes are linear systems and the coupling is linear and diffusive, that is when

$$\dot{x}_i = A(t)x_i + \Gamma \sum_{j \in \mathcal{N}_i} (x_j - x_i) \quad i = 1, \dots, N \quad (12.13)$$

where $A(t)$ is the dynamical matrix of each node and Γ is some inner coupling matrix, it is possible to prove the following theorem. (We include here a proof for the sake of completeness although other proofs of the same result are also available, see for example [12] for an overview.)

Theorem 12.7 *Network (12.13) synchronizes if there exist some matrix measure μ and constant $c > 0$ such that*

$$\mu(A(t) - \lambda_2 \Gamma) \leq -c \quad \forall t \geq 0$$

Proof Let J be the Jacobian of (12.13) given as

$$J(t) = I_N \otimes A(t) - L \otimes \Gamma,$$

and V be the orthonormal matrix spanning \mathcal{S}^\perp as in (12.11). Following Theorem 12.4 network (12.13) synchronizes if

$$\mu \left(V(I_N \otimes A(t) - L \otimes \Gamma)V^T \right) \leq -c,$$

that is

$$\mu \left(V(I_N \otimes A(t))V^T - V(L \otimes \Gamma)V^T \right) \leq -c \quad (12.14)$$

Now using (12.11) the first term in (12.14) can be recast as

$$\begin{aligned} V(I_N \otimes A(t))V^T &= (Q \otimes I_N)^T (I_N \otimes A(t))(Q \otimes I_N) = \\ &= (Q^T \otimes I_N)(I_N \otimes A(t))(Q \otimes I_N) = \\ &= (Q^T \otimes A(t))(Q \otimes I_N) = \\ &= I_N \otimes A(t) \end{aligned}$$

On the other hand, the second term of (12.14) can be written as

$$\begin{aligned} V(L \otimes \Gamma)V^T &= (Q \otimes I_n)^T (L \otimes \Gamma)(Q \otimes I_n) = \\ &= (Q^T \otimes I_n)(L \otimes \Gamma)(Q \otimes I_n) = \\ &= (Q^T L \otimes \Gamma)(Q \otimes I_n) = \\ &= Q^T L Q \otimes \Gamma = \\ &= \Lambda \otimes \Gamma \end{aligned}$$

This means that (12.14) become

$$\mu (I_N \otimes A(t) - \Lambda \otimes \Gamma) \leq -c.$$

Since $I_N \otimes A(t)$ and $\Lambda \otimes \Gamma$ are block diagonal matrices, we have

$$\mu (I_N \otimes A(t) - \Lambda \otimes \Gamma) = \mu (A(t) - \lambda_2 \Gamma) \leq -c,$$

proving the result. \square

As is the case for the MSF approach, these results nicely link the contraction properties of the node vector fields with the structural properties of the network and the strength of the coupling.

12.6.1 Example

To illustrate how contraction to a flow-invariant subspace can be successfully applied to networked dynamical systems, we consider the directed network of four FitzHugh-Nagumo neurons whose structure is depicted in Fig. 12.3a.

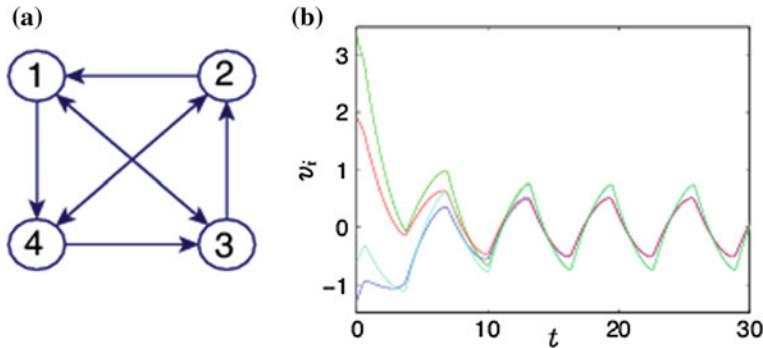


Fig. 12.3 The network structure of the example 12.6.1 (left panel). Simulation of the network (12.16) with parameters $a = 0$, $b = 2$, $c = 6$, coupling gains $\sigma_1 = 7$, $\sigma_2 = 10$, $k_1 = 1$, $k_2 = 10$, and $u(t) = 3 \sin(t)$. The initial conditions are uniformly distributed in the interval $[-4, 4]$ (right panel)

A single FN neuron can be described as [38]

$$\begin{cases} \dot{v} = f(v, w) = c \left(v + w - \frac{1}{3} v^3 + u(t) \right) \\ \dot{w} = g(v, w) = -\frac{1}{c}(v - a + bw) \end{cases} \quad (12.15)$$

where v is the membrane potential, w is a recovery variable, $u(t)$ is the magnitude of the stimulus current assumed to be periodic. The parameters a , b and c are non negative and they are set to $a = 0$, $b = 2$, $c = 6$.

In particular we want to prove that a subgroup of system nodes converges towards a specific manifold. The mathematical model of the network is

$$\begin{aligned} \dot{v}_1 &= f(v_1, w_1) + k_1 v_2 + \sigma_1(v_3 - v_1) \\ \dot{w}_1 &= g(v_1, w_1) \\ \dot{v}_2 &= f(v_2, w_2) + k_2 v_3 + \sigma_2(v_4 - v_2) \\ \dot{w}_2 &= g(v_2, w_2) \\ \dot{v}_3 &= f(v_3, w_3) + k_1 v_4 + \sigma_1(v_1 - v_3) \\ \dot{w}_3 &= g(v_3, w_3) \\ \dot{v}_4 &= f(v_4, w_4) + k_2 v_1 + \sigma_2(v_2 - v_4) \\ \dot{w}_4 &= g(v_4, w_4) \end{aligned} \quad (12.16)$$

As a representative example assume we want to prove convergence between nodes 1 and 3, that is we want to prove convergence of the network towards the manifold

$$\mathcal{M} := \left\{ [(v_1, w_1)^T, \dots, (v_4, w_4)^T] \in \mathbb{R}^8 : (v_1, w_1)^T = (v_3, w_3)^T \right\}$$

and therefore contraction on \mathcal{M}^\perp , whose basis is

$$V = [I_2 \ O_2 \ -I_2 \ O_2]$$

where I_2 and O_2 are respectively 2×2 identity and zero matrices.

The system Jacobian is

$$J = \begin{bmatrix} J_1 - \Sigma_1 & K_1 & \Sigma_1 & O_2 \\ O_2 & J_2 - \Sigma_2 & K_2 & \Sigma_2 \\ \Sigma_1 & O_2 & J_3 - \Sigma_1 & K_1 \\ K_2 & \Sigma_2 & O_2 & J_4 - \Sigma_2 \end{bmatrix}$$

where

$$J_i = \begin{bmatrix} \frac{\partial f}{\partial v_i} & \frac{\partial f}{\partial w_i} \\ \frac{\partial g}{\partial v_i} & \frac{\partial g}{\partial w_i} \end{bmatrix} = \begin{bmatrix} c(1 - v_i^2) & c \\ -\frac{1}{c} & -\frac{b}{c} \end{bmatrix}$$

and

$$\Sigma_i = \begin{bmatrix} \sigma_i & 0 \\ 0 & 0 \end{bmatrix}, \quad K_i = \begin{bmatrix} k_i & 0 \\ 0 & 0 \end{bmatrix}.$$

To prove contraction towards \mathcal{M}^\perp Theorem 12.4 can be applied, therefore we need to study the matrix measure μ of

$$VJV^T = J_1 + J_3 - 4\Sigma_1 = \begin{bmatrix} \frac{\partial f}{\partial v_1} + \frac{\partial f}{\partial v_3} - 4\sigma_1 & 2c \\ -\frac{2}{c} & -\frac{2b}{c} \end{bmatrix}$$

Using the ∞ -vector norm we get the condition

$$\mu_\infty(VJV^T) = \max \left\{ \frac{\partial f}{\partial v_1} + \frac{\partial f}{\partial v_3} - 4\sigma_1 + |2c|; -\frac{2b}{c} + \left| -\frac{2}{c} \right| \right\} < 0 \quad (12.17)$$

Since $\frac{\partial f}{\partial v_i} = c(1 - v_i^2) < c \quad \forall v_i$, the first term in (12.17) is negative definite if

$$c + c - 4\sigma_1 < -2c \implies \sigma_1 > c$$

while the second term is negative if $b > 1$.

Therefore condition (12.17) is fulfilled if

$$\begin{cases} \sigma_1 > c \\ b > 1 \end{cases}$$

A numerical simulation is reported in Fig. 12.3b showing that nodes 1 and 3 synchronize. Furthermore, nodes 2 and 4 synchronize as well on a different solution in \mathcal{M} , and the network achieves the so-called *cluster synchronization* regime [15, 39].

12.7 Hierarchical Approach

A final alternative which is particularly useful in the case of large networks of interconnected systems is to use a hierarchical approach to obtain a conservative estimate of the matrix measure of the Jacobian of the system or network of interest using multiple norms [19]. Specifically, contraction of the overall system can be guaranteed if some matrix measure of the Jacobian of each individual node is upper bounded and the measure of a reduced-order matrix associated to the interconnection structure is negative.

Let us consider two norms, a *local* norm $|\cdot|_L$ on \mathbb{R}^n and a *structure* norm $|\cdot|_S$ on \mathbb{R}^N assumed to be monotone. Given any vector $X = [x_1^T, \dots, x_n^T]^T \in \mathbb{R}^{nN}$, with $x_i \in \mathbb{R}^n$, $i = 1, \dots, N$, we define a *global* norm on \mathbb{R}^{nN} as

$$|X|_G := \left\| \begin{bmatrix} |x_1|_L, \dots, |x_N|_L \end{bmatrix}^T \right\|_S. \quad (12.18)$$

Furthermore we use the notations $\mu_L(\cdot)$, $\mu_S(\cdot)$ and $\mu_G(\cdot)$ to denote the matrix measures associated to $\|\cdot\|_L$, $\|\cdot\|_S$ and $\|\cdot\|_G$ respectively.

Let $J(x, t)$ denote the $nN \times nN$ Jacobian matrix of (12.4). Consider the following partition

$$J(x, t) = \begin{bmatrix} J_{11} & J_{12} & \dots & J_{1k} \\ J_{21} & J_{22} & \dots & J_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ J_{k1} & J_{k2} & \dots & J_{kk} \end{bmatrix}$$

and define the $k \times k$ *structure Jacobian* matrix as

$$J_S(x, t) = \begin{bmatrix} \tilde{J}_{11} & \tilde{J}_{12} & \dots & \tilde{J}_{1k} \\ \tilde{J}_{21} & \tilde{J}_{22} & \dots & \tilde{J}_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{J}_{k1} & \tilde{J}_{k2} & \dots & \tilde{J}_{kk} \end{bmatrix} \quad (12.19)$$

with

$$\begin{cases} \tilde{J}_{ii} = \mu_L(J_{ii}(x, t)) & \text{for } i \in \{1, \dots, k\} \\ \tilde{J}_{ij} = \|J_{ij}(x, t)\|_L & \text{for } i, j \in \{1, \dots, k\}, i \neq j \end{cases}$$

The following result holds (see [19] for a proof).

Theorem 12.8 *For every local norm on \mathbb{R}^n , every monotone structure norm on \mathbb{R}^N and every matrix $J \in \mathbb{R}^{nN \times nN}$*

$$\mu_G(J) \leq \mu_S(J_S) \quad (12.20)$$

Therefore, let \mathcal{C} be a convex set in \mathbb{R}^{nN} , if

$$\mu_S(J_S(x, t)) \leq -c \quad \forall x \in \mathcal{C}, \forall t \geq 0$$

then system (12.4) is contracting in \mathcal{C} .

Since (12.20) is conservative, the analysis is robust in the sense that a large degree of uncertainty can be tolerated in the components as long as the estimations are met for the network subsystems and their couplings. (A full proof of convergence together with further details and a generalization to nonidentical nodes can be found in [19, 40].)

The hierarchical approach described above can be effectively used to prove that a system or network is contracting. In the latter case, if the synchronization manifold is embedded into the region where the network is contracting, since it is an invariant set all trajectories will converge towards it and synchronization will remain proved. More generally, the hierarchical approach can reduce the problem of evaluating the matrix measure of the Jacobian of the virtual system or that of the matrix used when proving contraction to a manifold. Hence, it is a useful tool when proving synchronization of large networks with a well identified hierarchical structure.

12.7.1 Example

We illustrate how the hierarchical approach can be used to give sufficient conditions for the convergence of networks of nonlinear agents and hence as an effective design tool to select appropriate coupling functions. We focus again on the representative network of FitzHugh-Nagumo neurons (12.15) which was also studied in [19].

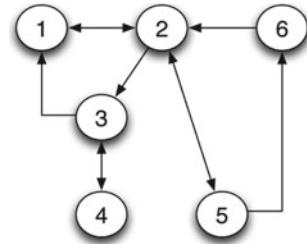
The key steps of this methodology can be summarised as follows:

1. we first compute the reduced-order structured matrix J_S in (12.19);
2. we then use such reduced-order matrix to design appropriate coupling protocols between the network nodes in order to guarantee network convergence.

The network topology considered in this example is shown in Fig. 12.4.

We assume the coupling protocol among nodes to be similar to the so-called *excitatory-only coupling*, specifically the FN oscillators are coupled via the coupling function

Fig. 12.4 The network used for the simulation of FN neurons (Reproduced from Fig. 12.1 in [19], with permission.)



$$\tilde{h}_i(v_i, w_i) := -\gamma_1 \sum_{j \in \mathcal{N}_i} v_j - (\gamma_2 + c)v_i, \quad \gamma_1, \gamma_2 > 0 \quad (12.21)$$

which is added to the first state equation in (12.15). Thus, the network dynamics become

$$\begin{cases} \dot{v}_i = c(v_i + w_i - \frac{1}{3}v_i^3 + u(t)) + \tilde{h}_i(v_i, w_i) \\ \dot{w}_i = -\frac{1}{c}(v_i - a + bw_i) \end{cases} \quad i = 1, \dots, N \quad (12.22)$$

To guarantee that all nodes converge towards a unique trajectory in state space, it now suffices to choose the parameters of the coupling protocol (12.21) so as to make the closed-loop network (12.22) contracting. Indeed, the synchronization manifold (12.7), with $x_i = [v_i, w_i]^T$, is flow invariant for the network dynamics (that is, trajectories with initial conditions in \mathcal{S} remain in it for all $t \geq t_0$). Hence, if the network is contracting, trajectories starting from any two initial conditions will converge exponentially towards each other. As trajectories in \mathcal{S} remain therein for all time, it immediately follows that all trajectories must converge towards \mathcal{S} and asymptotically towards each other; that is, nodes will synchronize. Moreover, from Theorem 12.3, network contraction also yields that the synchronous evolution will be periodic with the same period of the external stimulus $u(t)$.

Now, to study contraction of the network dynamics we would need to study the Jacobian of (12.22), which is a $2N \times 2N$ matrix. Using the hierarchical approach, we can look instead at the structure Jacobian, which is, in this case, an $N \times N$ matrix, defined in (12.19). To compute J_S we use as *local* norm on \mathbb{R}^2 the one induced by the ∞ -norm, and choosing $\gamma_2 > c + b - 1/c$ we have

$$J_S = \begin{bmatrix} -\frac{b-1}{c} & \gamma_1 & \gamma_1 & 0 & 0 & 0 \\ \gamma_1 & -\frac{b-1}{c} & 0 & 0 & \gamma_1 & \gamma_1 \\ 0 & \gamma_1 & -\frac{b-1}{c} & \gamma_1 & 0 & 0 \\ 0 & 0 & \gamma_1 & -\frac{b-1}{c} & 0 & 0 \\ 0 & \gamma_1 & 0 & 0 & -\frac{b-1}{c} & 0 \\ 0 & 0 & 0 & 0 & \gamma_1 & -\frac{b-1}{c} \end{bmatrix}$$

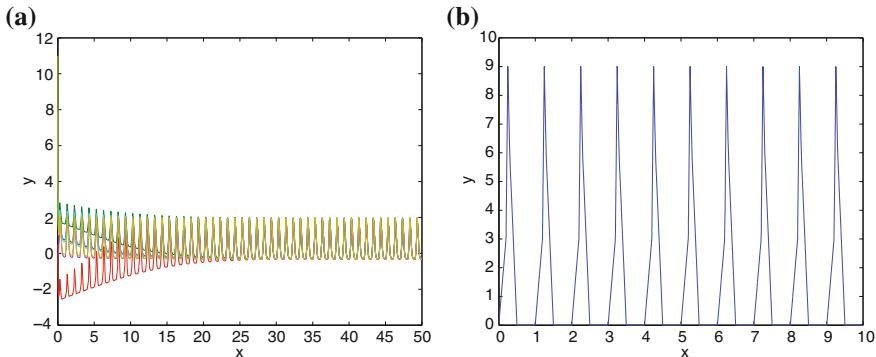


Fig. 12.5 Simulation of (12.22) with $\gamma_1 = 0.05$ and $\gamma_2 = 7$ (left panel). The behaviour of $u(t)$ is shown ($0 \leq t \leq 10$) in the right panel (Reproduced from Fig. 12.1 in [19], with permission.)

Now to ensure network contraction (and hence synchronization) we have to tune the parameters b and γ_1 so that a uniformly negative structure matrix measure in \mathbb{R}^6 for J_S exists.

Choosing as *structure* norm the 1-vector norm, we have that the matrix measure $\mu_1(J_S)$ is uniformly negative if

$$\begin{aligned} \{J_S\}_{ii} &< 0 \\ |\{J_S\}_{ii}| &> \sum_{i \neq j} |\{J_S\}_{ij}| \end{aligned}$$

Notice that the first condition above is fulfilled if $b > 1$. The second condition is instead guaranteed if

$$3\gamma_1 < \frac{b-1}{c}$$

Thus if this last condition is fulfilled, all network trajectories will converge towards a unique synchronous solution. A simulation for such a network is reported in Fig. 12.5, confirming the theoretical predictions.

12.8 Recent Extensions

All the results presented so far can be applied to the case when the node vector fields are sufficiently smooth and differentiable. Also the network structure is assumed to be time-invariant. As reported in the literature this is not the case in many applications where the network structure changes in time and the nodes are described by non-differentiable vector fields. Take for example the case of networks of power electronic converters in Electronics [41], friction oscillators in Earthquake engineering [42]

or all those examples whose dynamics is affected by discontinuous events on a macroscopic timescale [43].

Recently, an extension of contraction analysis to some classes of switched systems has been presented that can be effectively used to prove synchronization in these classes of systems. Specifically, contraction of Carathéodory systems was discussed in [44]. It was shown that the switched system of interest is contracting if a unique matrix measure exists such that each of the system modes is contracting. This result was then relaxed in [45] where multiple matrix norms can be used, each for a different system configuration. Contraction of switched systems exhibiting sliding mode solutions was discussed in [46] in the planar case and extended in [47] to n -dimensional systems.

As a representative example, we consider a network of time-switched systems of the form [44]

$$\dot{x}_i = \begin{bmatrix} 0 & \text{saw}_T(t) \\ -1 & 0 \end{bmatrix} x_i + \Gamma \sum_{j \in \mathcal{N}_i} (x_j - x_i)$$

where $x_i \in \mathbb{R}^2$, $\text{saw}_T(t) : \mathbb{R} \rightarrow [0, 1]$ is a saw-tooth wave of period T , Γ is a coupling matrix, often termed as inner-coupling matrix in the literature, chosen as

$$\Gamma = k \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

with k being the coupling gain to be determinate to attain synchronization. The network considered here consists of an all-to-all topology of three nodes, hence the Laplacian matrix is given as

$$L = \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix}$$

and $\lambda_2 = 3$, where λ_2 is the smallest non-zero Laplacian eigenvalue, the so-called algebraic connectivity. Thus, from Theorem 5.1 in [44] it follow that the network synchronizes if there exist some matrix measure and a positive scalar c such that

$$\mu(A(t) - \lambda_2 \Gamma) \leq c, \quad c > 0$$

that is if

$$\mu \left(\begin{bmatrix} 0 & \text{saw}_T(t) \\ -1 & 0 \end{bmatrix} - 3k \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right) = \mu \left(\begin{bmatrix} -3k & \text{saw}_T(t) \\ -1 & -3k \end{bmatrix} \right) \leq -c$$

Now, using the matrix measure induced by the 1-vector norm (column sums) and considering the fact that $\text{saw}_T(t)$ is a bounded signal with $\text{saw}_T(t) \leq 1$, it is

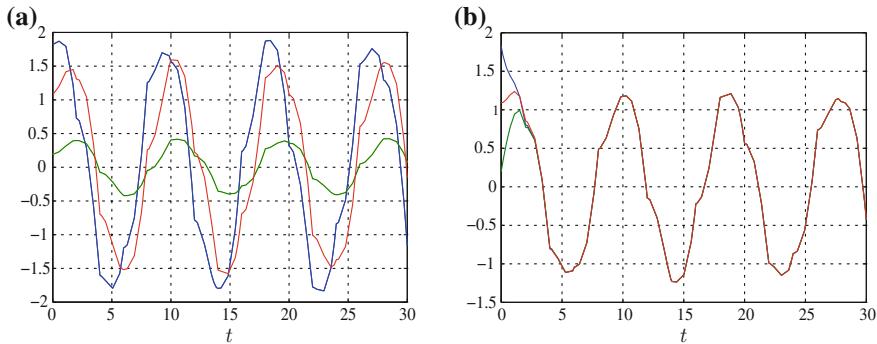


Fig. 12.6 Time evolution of the first state components of the networked switched linear system: (a) $k = 0$, (b) $k = 0.4$ (Reproduced from Fig. 5.1 in [44]. Copyright © 2014 Society for Industrial and Applied Mathematics. Reprinted with permission. All rights reserved.)

straightforward to see that the above conditions are fulfilled if the coupling gain is selected as

$$k > \frac{1}{3}.$$

As shown in Fig. 12.6, the theoretical predictions are confirmed by the numerical simulations obtained considering $T = 2$ and random initial conditions.

12.9 Conclusions

We have discussed different approaches to study convergence and synchronization of networks using contraction analysis. After introducing some background and an overview of the key results, we focussed on three complementary methods to prove convergence in networks of nonlinear systems. Firstly, the technique based on the construction of a virtual system was presented that associates to the network of interest a system whose particular solutions are those of its nodes. Contraction of the virtual system can then be used to prove synchronization. A second possibility is to study convergence of all node trajectories towards the synchronization subspace using tools to prove contraction of a system towards an invariant manifold. As a final approach, we studied the case of hierarchical structures showing that contraction can be effectively proved if the Jacobian of the system or network of interest can be appropriately partitioned. In all cases, representative examples were used to illustrate the effectiveness of the proposed strategy. Finally, recent extensions were discussed to the case where the node dynamics or network structure are affected by discontinuities.

References

1. Cao, Y., Yu, W., Ren, W., Chen, G.: An overview of recent progress in the study of distributed multi-agent coordination. *IEEE Trans. Ind. Inf.* **9**(1), 427–438 (2013)
2. Cornelius, S.P., Kath, W.L., Motter, A.E.: Realistic control of network dynamics. *Nat. Commun.* **4**(1942) (2013)
3. Liu, Y., Slotine, J., Barabasi, A.L.: Controllability of complex networks. *Nature* **473**(7346), 167–173 (2011)
4. Pecora, L.M., Carroll, T.L.: Master stability functions for synchronized coupled systems. *Phys. Rev. Lett.* **80**(10), 2019–2112 (1998)
5. Li, Z., Chen, G.: Global synchronization and asymptotic stability of complex dynamical networks. *IEEE Trans. Circuits Syst. II: Express Briefs* **53**(1), 28–33 (2006)
6. McMillen, D., Kopell, N., Hasty, J., Collins, J.J.: Synchronizing genetic relaxation oscillators by intercell signaling. *Proc. Natl. Acad. Sci.* **99**(2), 679–684 (2002)
7. Lohmiller, W., Slotine, J.J.E.: On contraction analysis for non-linear systems. *Automatica* **34**(6), 683–696 (1998)
8. Russo, G., di Bernardo, M., Slotine, J.J.: A graphical algorithm to prove contraction of nonlinear circuits and systems. *IEEE Trans. Circuits Syst.* **58**(2), 336–348 (2011)
9. Russo, G., di Bernardo, M., Sontag, E.D.: Global entrainment of transcriptional systems to periodic inputs. *PLoS Comput. Biol.* **6**(4), e1000739 (2010)
10. Forni, F., Sepulchre, R.: A differential Lyapunov framework for contraction analysis. *IEEE Trans. Autom. Control* **59**(3), 614–628 (2014)
11. Simpson-Porco, J.W., Bullo, F.: Contraction theory on Riemannian manifolds. *Syst. Control Lett.* **65**, 74–80 (2014)
12. Aminzare, Z., Sontag, E.D.: Contraction methods for nonlinear systems: a brief introduction and some open problems. In: Proceedings of IEEE Conference on Decision and Control, pp. 3835–3847 (2014)
13. Aminzare, Z., Sontag, E.D.: Remarks on diffusive-link synchronization using non-Hilbert logarithmic norms. In: Proceedings of IEEE Conference on Decision and Control, pp. 6086–6091 (2014)
14. Wang, W., Slotine, J.J.: On partial contraction analysis for coupled nonlinear oscillators. *Biol. cybern.* **92**(1), 38–53 (2005)
15. Pham, Q.C., Slotine, J.J.: Stable concurrent synchronization in dynamic system networks. *Neural Netw.* **20**(1), 62–77 (2007)
16. Russo, G., di Bernardo, M.: How to synchronize biological clocks. *J. Comput. Biol.* **16**(2), 379–393 (2009)
17. Russo, G., di Bernardo, M.: Solving the rendezvous problem for multi-agent systems using contraction theory. In: Proceedings of the IEEE Conference on Decision and Control, pp. 5821–5826 (2009)
18. Russo, G., Slotine, J.J.E.: Global convergence of quorum-sensing networks. *Phys. Rev. E* **82**, 041919 (2010)
19. Russo, G., di Bernardo, M., Sontag, E.D.: A contraction approach to the hierarchical analysis and design of networked systems. *IEEE Trans. Autom. Control* **58**(5), 1328–1331 (2013)
20. Hartman, P.: On stability in the large for systems of ordinary differential equations. *Can. J. Math.* **13**(3), 480–492 (1961)
21. Lewis, D.C.: Metric properties of differential equations. *Am. J. Math.* **71**(2), 294–312 (1949)
22. Pavlov, A., Pogromsky, A., van de Wouw, N., Nijmeijer, H.: Convergent dynamics, a tribute to Boris Pavlovich Demidovich. *Syst. Control Lett.* **52**(3), 257–261 (2004)
23. Angeli, D.: A Lyapunov approach to incremental stability properties. *IEEE Trans. Autom. Control* **47**(3), 410–421 (2002)
24. Lohmiller, W., Slotine, J.J.E.: Contraction analysis of non-linear distributed systems. *Int. J. Control* **78**(9), 678–688 (2005)
25. Slotine, J.J., Li, W.: *Applied Nonlinear Control*. Prentice Hall, Englewood Cliffs (1990)

26. Bertsekas, D., Tsitsiklis, J.: *Parallel and Distributed Computation: Numerical Methods*. Prentice-Hall, Englewood Cliffs (1989)
27. Jouffroy, J.: Some ancestors of contraction analysis. In: Proceedings of the IEEE Conference on Decision and Control, pp. 5450–5455 (2005)
28. Vidyasagar, M.: *Nonlinear Systems Analysis*. SIAM, Philadelphia (2002)
29. Khalil, H.K.: *Nonlinear Systems*. Prentice Hall, Upper Saddle River (2002)
30. DeLellis, P., di Bernardo, M., Gorochowski, T.E., Russo, G.: Synchronization and control of complex networks via contraction, adaptation and evolution. *Circuits Syst. Mag. IEEE* **10**(3), 64–82 (2010)
31. Dorfler, F., Bullo, F.: Synchronization in complex networks of phase oscillators: a survey. *Automatica* **50**(6), 1539–1564 (2014)
32. Olfati-Saber, R., Fax, J.A., Murray, R.M.: Consensus and cooperation in networked multi-agent systems. *Proc. IEEE* **95**(1), 215–233 (2007)
33. Ren, W., Cao, Y.: *Distributed Coordination of Multi-agent Networks*. Springer, London (2011)
34. Elowitz, M.B., Leibler, S.: A synthetic oscillatory network of transcriptional regulators. *Nature* **403**(6767), 335–338 (2000)
35. Garcia-Ojalvo, J., Elowitz, M.B., Strogatz, S.H.: Modeling a synthetic multicellular clock: repressilators coupled by quorum sensing. *Proc. Natl. Acad. Sci. USA* **101**(30), 10955–10960 (2004)
36. Russo, G., Di Bernardo, M.: Contraction theory and master stability function: linking two approaches to study synchronization of complex networks. *IEEE Trans. Circuits Syst.* **56**(2), 177–181 (2009)
37. Russo, G., Di Bernardo, M.: Design of coupling protocols for consensus and cluster synchronization using contraction theory (In preparation)
38. FitzHugh, R.: Mathematical models of threshold phenomena in the nerve membrane. *Bull. Math. Biophys.* **17**(4), 257–278 (1955)
39. Golubitsky, M., Stewart, I., Török, A.: Patterns of synchrony in coupled cell networks with multiple arrows. *SIAM J. Appl. Dyn. Syst.* **4**(1), 78–100 (2005)
40. Russo, G., di Bernardo, M., Sontag, E.D.: Stability of networked systems: a multi-scale approach using contraction. In: Proceedings of IEEE Conference on Decision and Control, pp 6559–6564 (2010)
41. Karimi-Ghartemani, M., Iravani, M.R.: A method for synchronization of power electronic converters in polluted and variable-frequency environments. *IEEE Trans. Power Syst.* **19**(3), 1263–1270 (2004)
42. Chopra, A.K.: *Dynamics of Structures*. Prentice Hall, Upper Saddle River (2011)
43. di Bernardo, M., Budd, C., Champneys, A.R., Kowalczyk, P.: *Piecewise-Smooth Dynamical Systems: Theory and Applications*, vol. 163. Springer Science & Business Media, London (2008)
44. di Bernardo, M., Liuzza, D., Russo, G.: Contraction analysis for a class of nondifferentiable systems with applications to stability and network synchronization. *SIAM J. Control Optim.* **52**(5), 3203–3227 (2014)
45. Lu, W., di Bernardo, M.: Contraction and incremental stability of switched Carathéodory systems using multiple norms (Submitted to Automatica)
46. di Bernardo, M., Liuzza, D.: Incremental stability of planar Filippov systems. In: Proceedings of the IEEE European Control Conference, pp 3706–3711 (2013)
47. di Bernardo, M., Fiore, D.: Incremental stability of bimodal Filippov systems in \mathbb{R}^n . In: Proceedings of IEEE Conference on Decision and Control, pp 4679–4684 (2014)

Chapter 13

Towards Structural Controllability of Temporal Complex Networks

Xiang Li, Peng Yao and Yujian Pan

Abstract Temporal complex networks are ubiquitous in human daily life whose topologies evolve with time, such as communication networks and transportation networks. Investigations on the structural controllability of temporal complex networks show the properties and performances of controllability when the weights of edges in temporal networks are arbitrary values rather than exact ones. There are two frameworks proposed in this chapter to analyze the structural controllability of temporal networks. In the first framework, a temporal network is treated as a sequence of characteristic subgraphs with different characteristic time stamps. After finding the maximum characteristic subgraph set from these subgraphs, priority maximum methods are applied to improve the controlling efficiency by which temporal information of the network remains. On the other hand, in the later framework, a temporal network is represented by time-ordered graph (TOG). Instead of calculating the rank of controllability matrix directly, finding and classifying temporal trees of the time-ordered graph provides an effective way to estimate the controlling centrality of a node in the network.

13.1 Introduction

The past decades have witnessed flourishing advances of understanding the essence of complex networking worlds, where the seminar discoveries include the well-known “small-world” and “scale-free” models [2, 69] to capture different topological patterns of complex networks, ranging from the Internet, WWW, air-line transportation networks, to categories of social networks, economic networks, and biological and ecological networking systems [3, 4, 11, 15, 21, 45, 51, 57, 66, 71]. Such achievements not only help people show sufficient respect to the significance of connectivity

X. Li (✉) · P. Yao · Y.J. Pan

Adaptive Networks and Control Lab, Department of Electronic Engineering,
and Research Center of Smart Networks & Systems, School of Information
Science & Engineering, Fudan University, Shanghai 200433,
People’s Republic of China
e-mail: lix@fudan.edu.cn

patterns of complex networks to their collective behaviours and functional performance, but also arouse wide attention to the huge desire of controlling complex networks. The fruitful outcomes to pinning a small fraction of nodes in a network to achieve the desired collective behaviours, i.e., the so-called pinning control [7, 17, 33, 40, 58–60, 65, 67], have exhibited the feasibility and effectiveness of controlling complex networks with the latest extension to multi-agent networks [76, 77]. When neglecting the node dynamics embedded into a network and concerning reachability of control signals, structural controllability as well as strong structural controllability of a network is a key factor to depict the topological complexity of controlling a complex network [34, 36, 39]. Very recently, there are continuous efforts devoted to characterize structural controllability of complex networks, such as detecting control-inducing structures [8, 56], measuring control effect [24, 25, 37, 63], optimizing controllability [64, 72, 73], investigating robustness [9, 61, 68], measuring strong structural controllability [5, 23], etc [6, 10, 35, 38, 44, 46, 75].

Note that a stationary network does not exist in reality, where the connections among nodes evolve with time, no matter very fast or slowly even when we can not notice the change easily. More importantly, due to the explosion of digital technology and prevalence of electronic communication services, huge amounts of data in large scale networking systems are produced and recorded with time-stamps, which may come from face-to-face conversations in the era of network today, e-mail exchanges and cyber-interactions in various online behaviours of human population [12, 18, 22, 70, 78, 79, 82]. Such data are collectively described as temporal networks at specific time scales, where time-stamped events are edges between pairs of nodes [19]. Since nodes and (or) edges in a temporal network are only valid (or activated) at specific time stamps (or during specific time periods), such time sensitivity leads to various characteristics and collective behaviours on temporal networks, such as power-law distributions and bursts in contact intervals [1, 27], specific patterns in communication [14, 30], motifs in human interactions [78–82], etc [29, 31, 47, 52, 53, 55, 62].

As a natural raised significance outlined above, structural controllability on temporal networks deserves sufficient attention, which, however, has not been extensively visited as well. This chapter comes as a synthetical review of our continuous efforts devoted as the first step to exploring structural controllability of temporal complex networks with both theoretical and numerical supports of empirical temporal data sets [48–50]. In more detail, we first introduce as preliminary the definitions and concepts of structural controllability, followed by a brief overview of the productive researches in the line of complex networks with static connections. After we give the background and motivation of temporal networks, we mainly present two means of studying temporal networks: one is associating a temporal network with a sequence of linear time invariant (LTI) systems, and the other is describing a temporal network from the viewpoint of a linear time variant (LTV) system, based on which new maximum matching method and controlling centrality bounds are given. Finally, we summarize the whole chapter with discussions of future steps.

13.2 Structural Controllability of Complex Networks: Preliminary and Brief Review

13.2.1 Concepts and Definitions

Structural controllability was firstly proposed in [34], which concerns the topological controllability of a graph, and very recently presented to characterize the minimal driven nodes in a complex network [36]. To associate a static (i.e., the edges and nodes are fixed without evolving/switching) network with a linear time invariant (LTI) system [41], we have

$$\dot{x} = A'x(t) + Bu \quad (13.1)$$

where $A' \in R^{N \times N}$ denotes the transpose of the network adjacency matrix A , and N is the number of nodes in the network. $\forall a_{ij} \in A$, $a_{ij} \neq 0$ if there is an edge from node i to node j , and $a_{ij} = 0$, if there is no edge from node i to node j . Especially, for an indirected network, $a_{ij} = a_{ji}$. $x(t) = [x_1(t), x_2(t), \dots, x_N(t)]' \in R^N$ denotes the vector of nodes. $B \in R^{N \times M}$ denotes the input matrix, and M is the number of controllers on the network. $\forall b_{ij} \in B$, $b_{ij} \neq 0$ if there is controller j placed on node i , or $b_{ij} = 0$. $u(t) = [u_1(t), u_2(t), \dots, u_M(t)]' \in R^M$ denotes the vector of input signals from the external controllers on the network. It is well known that the Kalman's state controllability condition [26] tells that the LTI system presented by pair (A, B) is state controllable if and only if the controllability matrix $[(A')^{N-1}B, (A')^{N-2}B, \dots, (A')B, B]$ has full rank. Lin extended the state controllability to structural controllability in [34], which was applied to complex networks in [36] as:

Definition 13.1 A network presented by a pair (A, B) is structurally controllable, if there exists a state controllable system with the same structure (the non-zero elements in pair (A, B) have the same locations) with the network's associated system.

Definition 13.2 A network contains a dilation, if there is a subset $S_b \subset V$, and the set of its neighbour nodes $T(S_b)$ (edge starts from a node in S_b to a node in $T(S_b)$), where $|T(S_b)| > |S_b|$. V is the node set of the network.

A structural controllable network should have none inaccessible node and none dilation.

Definition 13.3 (a) In a network, a matching m is an edge subset where no two edges sharing a common starting vertex or a common ending vertex. The maximum set m^* is the maximum matching of the network. Denote $|m^*|$ the size of the maximum matching m^* .

(b) A node is matched if there is an edge in m points to it. Otherwise, the node is unmatched.

(c) The network has a perfect matching if there is no unmatched node to a maximum matching.

Liu et al. [36] successfully implemented the maximum matching algorithm [20] to estimate the minimum number of controllers and their locations (driven nodes) to ensure the structural controllability of complex networks by minimum inputs theorem [43]:

Theorem 13.1 *The minimum number of inputs to fully control a network, denoted as N_I , is equal to the minimum number of driver nodes, denoted as N_D . The number is equal to one if there is a perfect matching in a network, and any node can be the driven node. Otherwise, it is equal to the number of unmatched nodes to a maximum matchings, and these unmatched nodes are driven nodes.*

$$N_I = N_D = \max(N - |m^*|, 1) \quad (13.2)$$

13.2.2 Brief Review of Structural Controllability of Complex Networks

Stimulated by Liu et al.'s work on structural controllability of complex networks [36], wide attention has focused on structural role of complex networks on controllability as well as its extensions such as control energy, control profile, evolution, and the dual of observability. We will give a snapshot to review the fast emergent yet fruitful understanding of structural controllability of complex networks with static topological connectivity.

As a direct extension of the work [36], Liu et al. also investigated controlling centrality of a single node to characterize its influence to the whole network, and found that the contribution of a single node to controllability is determined by its hierarchy where the node locates [37], which is similarly measured in [63]. In [24, 25], Jia et al. divided all nodes into three types (critical, intermittent or redundant), i.e., the nodes must be, or may be, or can not be driven nodes depending on their probabilities of becoming driven nodes, and they found that such probabilities are negatively relative to the nodes' in-degrees and independent to their out-degrees [24]. Very recently, Ruths et al. found three control-inducing structures: source, internal-dilation, and external-dilation, which classify the whole network's control profile [56].

To optimize such structural controllability of a network, the efforts with minimal structural perturbations were discussed [68] by selectively adding edges to form a path containing all nodes, where the minimum number of driven nodes can reduce to one with remaining the network connectivity if such additional edges are available in practice. Similar interest comes to the concern that how much energy is required to achieve such a structural controllable network, and Yan et al. [72] claimed the upper and lower bounds of the control energy cost, which implies the tradeoff between control energy cost and control time. In this line, placing polar assignments to improve the closed-loop system performance of a structural controllable network deserves more systematic insight not limited to the initial trial [73].

Extending the “robustness and fragility” to structural controllability as other extensively visited topological metrics, to some extent, Wang et al. [64] investigated the robustness of structural controllability from the classical viewpoint of attack and failures. In a more realistic case, given the interactions between node states presenting nonlinear dynamics, the nodes’ perturbations may lead to the risk of network failure. Motter and his colleagues technically viewed such ill-conditioned situations and proposed their methods to ensure network controllability [9, 61].

State observability is the dual to state controllability in classic control theory of linear systems. Liu et al. developed network observability to estimate an internal node status via network outputs [38]. While the latest report has moved to the hybrid structure of network of networks (or interdependent networks), both controllability and observability [6] were involved with Cartesian Products, where the effectiveness of control and estimation were verified via a real example of network-of-networks.

More than structural controllability, strong structural controllability urges that the network (or system) should be controllable for any values of non-zero parameters in pair (A, B) [42]. Few yet notable works on strong structural controllability of linear systems refer to [23, 54]. Chapman and Mesbahi then proposed a constrained matching method to revisit strong structural controllability of a networking system [5]. Nevertheless, such strong controllability deserves further insight in a network with more general topology, whose structural controllability was claimed to precisely reachable via the Popov-Belevitch-Hautus (PBH) rank condition [75]. Before ending this quick survey, we should emphasize that structural controllability of a network can not always neglect the existence of nodal dynamics. Cowan et al. [10] pointed out that a controllable node should not only reach a desired state but also own the ability to maintain the status, and therefore nodal dynamics may come as the focal issue to determine structural controllability of a network, whatever static or dynamic evolving.

13.3 Temporal Network and Information Reachability

Before temporal network stands as the representative notation of a network with temporal information embedded, people have noticed the influence of time evolution on the connectivity of a network with the name such as time-evolving, time-switching, time-varying networks, etc. Traditionally, a time-switching network generally assumes its edges periodically switch to repeat some topological forms, and a time-varying network covers a general form of time-dependent topology which evolves continuously with time. Note that such understanding (or assumptions) of dynamical networks basically follows a poisson distribution of topological events as time evolves. When more and more non-poisson evidences observed, e.g., in many social activities such as face-to-face conversations, online interactions, and pairwise email communications obey power-law forms with burst features, the involved social networking connectivity patterns therein fall outside the traditional description of time-dependent networks [1, 27, 70]. Temporal networks therefore include rich

information on the duration and interval of two events represented by temporal edges, and their ordered time-sequence as well. More details may refer to the collections in [19, 81].

To visualize the significance of temporal networks as compared with previously wide concerned static networks, we only give an example on the information reachability with the illustration of two networks. In Fig. 13.1a, a temporal network is illustrated with 6 nodes and 7 edges, while such edges are only valid at some specific time stamps, which are indicated as the numbers in the parentheses over such temporal edges. As a comparison, Fig. 13.1b is a static version of Fig. 13.1a with the same nodes and edges, only removing the valid time stamps as the direct outcome of aggregation/projection from Fig. 13.1a, which is a traditional procedure before we give sufficient respect to the temporal dimension of a network. It is obvious that the network in Fig. 13.1b is connected, and every node is reachable from any other node. However, the case of temporal network in Fig. 13.1a is significantly different.

In Fig. 13.2a, we assume that at time $t = 0$, node A holds a piece of message to send. Since edge (A, B) is valid from $t = 1$ to $t = 4$, in Fig. 13.2b, when $t = 3$, message has arrived at node B via this edge. Then, in Fig. 13.2c, because of valid edges (C, D) and (E, F) , the message could reach node C and node F. However, if node D (or node E), for example, wants to get the message, there should exist at least a valid edge activated after $t = 4$ to receive the message from node C or node F, which fails as illustrated in Fig. 13.2d. Therefore, in the end, even though the message on node A can arrive at nodes B, C and F, it can never arrive at node D and E.

Such difference of information reachability in temporal networks as illustrated above also points out the significance of temporal structure to controllability of temporal networks, which obviously can not be regarded as a direct extension from the existed literature of structural controllability of (static) complex networks. In the remainder of this chapter, we mainly report our devoted efforts to explore structural controllability of temporal networks as reported in [48–50].

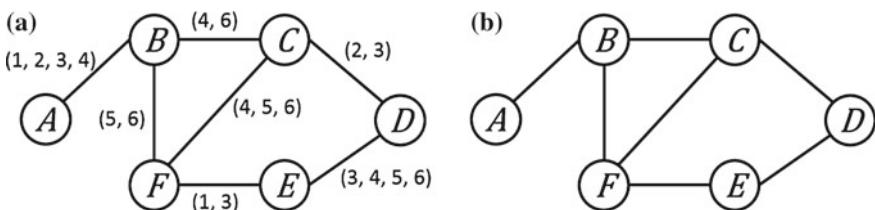


Fig. 13.1 An example of temporal network (a) and its corresponding static version (b). They both contain the same 6 nodes and 7 edges. However, in a, as temporal information included, edges are valid at some specific time stamps which are indicated by the numbers in those parentheses. In b, on the other hand, as a static network, edges are activated continuously.

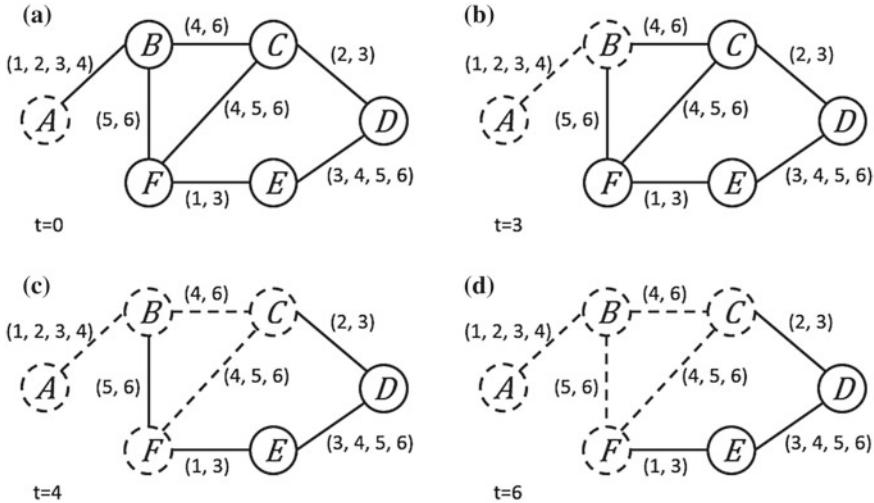


Fig. 13.2 [50] The illustration of information reachability of a temporal network at **a** $t = 0$, **b** $t = 3$, **c** $t = 4$ and **d** $t = 6$. Message sourced from node A could reach node B, C, F but could not reach node D, E for the absent edges (C, D) and (E, F) after $t = 4$. Therefore, in **(d)**, at $t = 6$, dashed circled node A, B, C, F are nodes holding the message, while node D, E couldn't get the message

13.4 Priority Maximum Matching Method on Temporal Networks

Denote triple (i, j, t) an edge of a temporal network between node i and node j activated at time t . We have the following definitions.

Definition 13.4 (a) Denote $T_C = \{t_1, t_2, \dots, t_k\}$, $t_i < t_{i+1}$, $i = 1, 2, \dots, k - 1$, the characteristic time sequence as the set of all the specific time stamps when edge (i, j, t) is activated.

(b) Denote $I_C = \{I_1, I_2, \dots, I_{k-1}\}$, the interval sequence as the set in which $I_i = t_{i+1} - t_i$, $i = 1, 2, \dots, k - 1$.

For example, as the temporal network in Fig. 13.2a, $T_C = \{1, 2, 3, 4, 5, 6\}$ and $I_C = \{1, 1, 1, 1, 1\}$ are the characteristic time sequence and the interval sequence, respectively.

The elements of interval sequence I_C include the smallest time-window width from the characteristic time sequence T_C , during which the network topology is assumed to be fixed. Therefore, for each interval of I_C , we have a “snapshot” of the temporal network as its characteristic sub-graph, from which we extract the maximum characteristic sub-graph set.

Figure 13.3 visualizes the process of transforming the temporal network in Fig. 13.1a into characteristic sub-graphs with its maximum characteristic sub-graph obtained. The contact sequence in Fig. 13.3b includes all the edges in Fig. 13.3a.

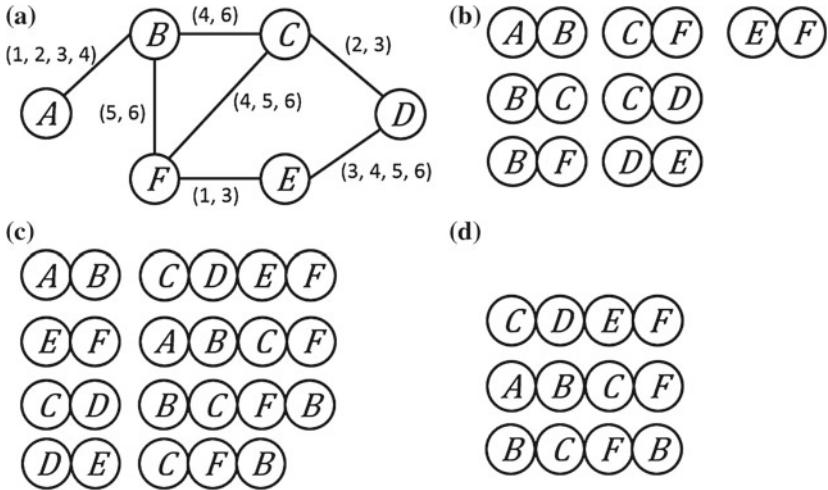


Fig. 13.3 Transforming a temporal network into sub-graphs. **a** The temporal network, **b** the contact sequence, **c** the set of characteristic sub-graphs during each time window, **d** the maximum characteristic sub-graph set

In each characteristic sub-graph, with common nodes, some edges may complete paths or circles which are included in Fig. 13.3c. For example, edges $(C, D, 3)$, $(D, E, 3)$, and $(E, F, 3)$ are all valid at $t = 3$, so a path from node C to F via D and E is completed and included in Fig. 13.3c. Depending on characteristic sub-graphs in (c), the maximum characteristic sub-graph set can be extracted and presented in Fig. 13.3d. For example, since $\{C, D, E, F\}$ contains $\{C, D\}$, $\{E, F\}$ and $\{D, E\}$, $\{C, D, E, F\}$ is included in Fig. 13.3d while $\{C, D\}$, $\{E, F\}$ or $\{D, E\}$ aren't.

In each characteristic sub-graph, we have the following definitions,

Definition 13.5 (a) A path with the odd number of nodes and the even number of edges is an odd path. An isolated node, which is not incident to any edge, is a special odd path.

(b) A path with the even number of nodes and the odd number of edges is an even path.

Definition 13.6 (a) A degeneration is a process in which an undirected graph is transformed into directed circles and isolated nodes by removing some edges. In a degeneration, an undirected edge can be treated as two directed edges in opposite directions with the same nodes.

(b) A perfect degeneration is a degeneration with the least isolated nodes.

We give an example of degeneration in Fig. 13.4 with an undirected network (Fig. 13.4a). As shown in Fig. 13.4b, by removing edges (A, B) , (B, G) , (B, E) and (C, D) , the undirected network leaves 3 isolated nodes (i.e., nodes A , D and G). While as shown in Fig. 13.4c, given removing edges (A, B) , (B, C) and (B, E) , the network only leaves 1 isolated node (i.e., node A). Since it is impossible to degenerate

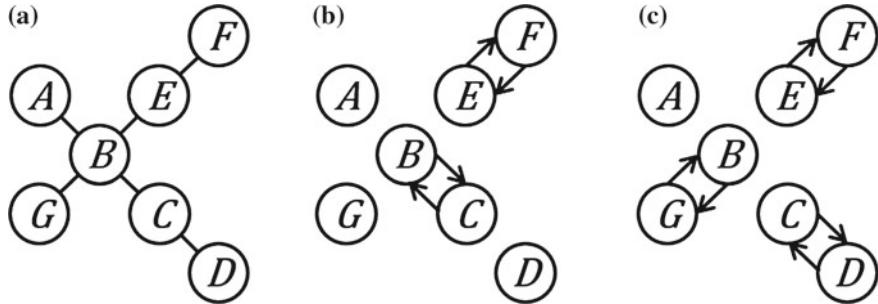


Fig. 13.4 Degeneration of an undirected network. **a** The origin network, **b** a degeneration with 3 isolated nodes, **c** a perfect degeneration with the least isolated node (1 node isolated)

the network with less isolated node left, the degeneration in Fig. 13.4c is a perfect degeneration.

Theorem 13.2 *The least number of nodes to structurally control a characteristic sub-graph (the minimum number of driven nodes) is $N_D = \max(|V_{isolated}|, 1)$, where $|V_{isolated}|$ denotes the number of isolated nodes in a perfect degeneration.*

Proof Two basic structures, path and circle, are considered in a matching (Definition 13.3), from [34] and [36]. The adjacency matrices of path and circle are

$$A_{n \times n}^p = \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 \\ a_1 & 0 & 0 & \cdots & 0 \\ 0 & a_2 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & a_{n-1} & 0 \end{pmatrix}, \text{ and } A_{n \times n}^c = \begin{pmatrix} 0 & 0 & 0 & \cdots & a_n \\ a_1 & 0 & 0 & \cdots & 0 \\ 0 & a_2 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & a_{n-1} & 0 \end{pmatrix}.$$

Especially, in a undirected graph, the adjacency matrix of a path is $A_{n \times n}^{p'} = \begin{pmatrix} 0 & a'_1 & 0 & \cdots & 0 \\ a_1 & 0 & a'_2 & \cdots & 0 \\ 0 & a_2 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & a'_{n-1} \\ 0 & 0 & \cdots & a_{n-1} & 0 \end{pmatrix}$.

Supposing that n is an even number (the path is an even path), $A_{n \times n}^{p'}$ can be degenerated into sub-matrices as $A_1^{p'} = \begin{pmatrix} 0 & a'_1 \\ a_1 & 0 \end{pmatrix}, \dots, A_{n/2}^{p'} = \begin{pmatrix} 0 & a'_{n-1} \\ a_{n-1} & 0 \end{pmatrix}$. And this is a perfect degeneration without any isolated nodes.

On the other hand, a perfect degeneration of an odd path has one isolated node and several circles. A maximum matching may contain several odd paths, so a perfect degeneration of the network may leave several isolated nodes. Since the isolated nodes are unmatched nodes in a maximum matching, by the minimum matching

theorem (Theorem 13.1), the minimum number of driven nodes in a sub-graph is equal to the number of isolated nodes, i.e., $N_D = \max(|V_{isolated}|, 1)$.

Specially, when the number of isolated nodes is equal to zero, the sub-graph is a perfect matching which implies that the minimum number of driven nodes is 1. \square

Given a temporal network, get the characteristic sub-graphs of the temporal network in each time window with width I_i , $i = 1, 2, \dots, k - 1$, respectively. Denote $S_C = \{S_{I_1}, S_{I_2}, \dots, S_{I_{k+1}}\}$ the set of these characteristic sub-graphs. Note that the edges in S_{I_i} , $i = 1, 2, \dots, k - 1$, are valid (activated) in I_i . According to S_C , the maximum characteristic sub-graph set is denoted as $S_M = \{S_j\}$, $S_j \in S_C$, and $\forall i \neq j$, S_j is not the subset of S_i . Then, get the priority sequence, $V_p = \{v_1, v_2, \dots, v_N\}$, where N is the number of nodes in the network. In the priority sequence, nodes are sorted in descending order by their frequency (times of a node presenting in all the characteristic sub-graphs) or influence (the number of reachable nodes that a node has). Choose nodes from the head of priority sequence V_p one by one as driven nodes.

To structurally control a temporal network, we propose the priority maximum matching method described as below:

1. If a characteristic sub-graph is a perfect matching, select the node with the highest priority in the priority sequence V_p as the driven node of the sub-graph. Otherwise, go to Step 2.
2. Removing nodes on the priority sequence V_p successively. The removed node is the driven node if the number of unmatched nodes in a maximum matching decreases. Otherwise, the node is not a driven node and restore it in the characteristic sub-graph. And remove the next node in the priority sequence until the number of selected driven nodes is equal to the number of unmatched nodes in a maximum matching of the original characteristic sub-graph.
3. For each characteristic sub-graph, do the same operations in Steps 1 and 2.

Definition 13.7 Define the Controlling Efficiency Index (CEI) to measure the control efficiency of a node as

$$CEI = \frac{NCO/NO}{NC/NDN} \quad (13.3)$$

where NCO denotes the number of controlled nodes, and NC denotes the number of driven nodes (not the minimum number of driven nodes), and it also denotes the number of controllers on the temporal network, and NO denotes the number of nodes in the network, and NDN denotes the number of unmatched nodes in a maximum matching (the number of isolated nodes in a perfect degeneration).

Note that the value of CEI ranges from zero to infinity, and a larger CEI indicates more control effectiveness with the limited involved controllers.

To apply the priority maximum matching method, we introduce several data sets to generate empirical temporal networks. “SG-Infectious” is the data set collected at the art-science gallery called “Infectious: stay away” in Dublin, Ireland, 2009, which

is accessible with the URL of <http://www.sociopatterns.org>. With the data set, nodes in the generated temporal network are the gallery attendees, and the edges represent the temporal contacts among the attendees. “Fudan WiFi” is the data set collected in 2009–2010 fall semester (3 months) at Fudan University, Shanghai, China, which is accessible with the URL of <http://www.can.fudan.edu.cn/data/fudanwifi09/>. In the temporal network generated from “Fudan WiFi”, nodes represent the WiFi users, and the users’ devices accessing the same AP (Access Point) during the same interval yield the temporal edges between the corresponding nodes in the network.

We further propose three priority strategies: “frequency” strategy means that the priority of node selection depends on the times of a node appearing in S_M , while “influence” strategy means that the priority ranks on the number of reachable nodes of a node in each sub-graph, and “randperm” strategy means that randomly shuffle the nodes for selection.

Figure 13.5 presents the outcomes of applying the three priority strategies to the generated temporal networks with the first 4 weeks of the above two empirical data sets, where two additional controllers are added on the network every step. As shown in Fig. 13.5, the “influence” strategy’s CEI is on average larger than that of the “frequency” strategy, while the “Randperm” strategy yields different performance with the data set of “SG-Infectious” (top) and “Fudan WiFi” (bottom), indicating the difference of the population behavioural patterns of two data sets. In more detail, “SG-Infectious” collected the attendee’s temporal behavioural information during the public exhibition. Since the attendee generally will not appear again after the visit, the temporal interactions among nodes in the generated temporal networks present more randomness than the periodic patterns of campus daily behaviours in “Fudan WiFi”.

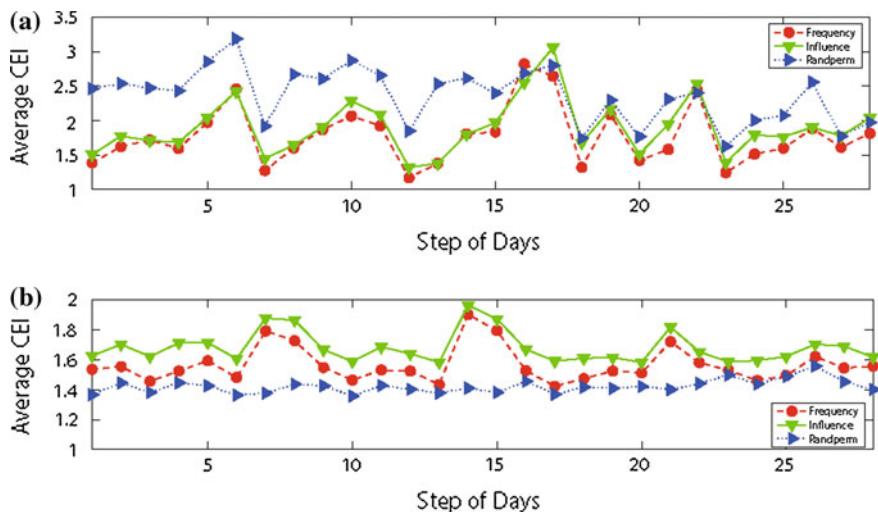


Fig. 13.5 [48] Average CEI of every day in the first 4 weeks: **a** SG-Infectious **b** Fudan WiFi. Curves labeled by ‘Frequency’, ‘Influence’ and ‘Randperm’ show the CEI applying strategies ‘Frequency’, ‘Influence’ and ‘shuffling randomly’, respectively

13.5 Node Controlling Centrality of Temporal Networks

To fully explore structural controllability of a temporal network, it is obvious that we can not keep satisfied to proxy a temporal network from the viewpoint of static networks. In the following section of this chapter, we will directly work on a temporal network with its associated linear-time-variant (LTV) system, given we agree with the line of linear-time-invariant (LTI) systems associated to static networks. As the first efforts to step into structural controllability of temporal networks, we assume one controller is fixed to drive the whole temporal network, and analytically estimate the network structural controllability with temporal information embedded.

13.5.1 Controlling Centrality

Following the line of associating a static network to a LTI system, we associate a temporal network to the LTV system as

$$\dot{x}(t) = A'(t)x(t) + B(t)u(t) \quad (13.4)$$

where $A'(t) \in R^{N \times N}$ denotes the transpose of the adjacency matrix of a temporal network, N denotes the number of nodes in the network, $x(t) \in R^N$ is the vector of the nodes, $B(t) \in R^{N \times M}$ is the input matrix, M denotes the number of controllers ($M = 1$ for a single controller), and $u(t) \in R^M$ is the input signal vector from controllers. Since we only consider the case of single fixed controller, the temporal network comes to

$$\dot{x}(t) = A'(t)x(t) + b^{(o)}u(t) \quad (13.5)$$

where $b^{(o)}$ denotes that the fixed controller is put on node i all the time. Embedding the intervals of temporal edges as the sampling sequence, we have the discrete form of Eq. (13.5) as

$$\frac{x(k+1) - x(k)}{T_{k+1}} = A'_{k+1}x(k) + b^{(o)}u(k) \quad (13.6)$$

where $T_{k+1}, k = 0, 1, \dots, T-1$ is the sequence of intervals in the order of time stamps of temporal edges, which works as sampling intervals. Note that it is not necessary to guarantee that $T_i = T_j$, for $i \leq j, 1 \leq i, j \leq T$, i.e., the sampling process is aperiodic.

Rewrite Eq. (13.6) as

$$x(k+1) = G_{k+1}x(k) + H_{k+1}u(k) \quad (13.7)$$

where $G_{k+1} = I + T_{k+1}A'_{k+1}$, $H_{k+1} = T_{k+1}b^{(o)}$, I is the identity matrix. Denote

$$W_c = [G_T \dots G_2 H_1, \dots, G_T H_{T-1}, H_T] \quad (13.8)$$

and the final state x_T is

$$\begin{aligned} x(T) &= [x_1(T), \dots, x_N(T)]' \\ &= [G_T \dots G_1] \cdot [x_1(0), \dots, x_N(0)]' \\ &\quad + W_c \cdot [u(0), u(1), \dots, u(T-1)]' \end{aligned} \quad (13.9)$$

Since x_T has two parts as

$$\begin{aligned} x(T) &= [x_1(T), \dots, x_k(T), x_{k+1}(T), \dots, x_N(T)]' \\ &= [G_T \dots G_1] \cdot [x_1(0), \dots, x_k(0), x_{k+1}(0), \dots, x_N(0)]' \\ &\quad + W_c \cdot [u(0), u(1), \dots, u(T-1)]' \end{aligned} \quad (13.10)$$

$\text{rank}(W_c) = k$ determines the controllable nodes of the network, while $\text{rank}(W_c) = N$ tells that the fixed controller on node o structurally control the whole temporal network. Therefore, we define the controlling centrality of node o given the fixed single controller on a temporal network:

Definition 13.8

$$S_{M(o)} = \text{rank}(W_c) \quad (13.11)$$

is the controlling centrality of node o .

When there is only a single controller fixed on a temporal network, the controlling centrality also equals to the maximum dimension of the controllable subspace. Due to the complexity of temporal networks, usually it is difficult to calculate $S_{M(o)}$ directly, and we estimate it with the assistance of time-ordered graph [28].

13.5.2 Time-Ordered Graph

Now it is time to give a formal notation of a temporal network, which, although, has been extensively discussed in this chapter. Denote $\mathcal{G}(V_{\mathcal{G}}, E_{\mathcal{G}})$ as a temporal network, where $V_{\mathcal{G}}$ and $E_{\mathcal{G}}$ are the sets of nodes and temporal edges, respectively. Denote $N(\mathcal{G}, T)$ the corresponding time-ordered graph of temporal network $\mathcal{G}(V_{\mathcal{G}}, E_{\mathcal{G}})$. Since it has been claimed that a temporal network can be transformed into a time-ordered graph with its temporal information remained [28], in the time-ordered graph as a directed graph, V_T is the node set of TOG including $T + 1$ duplications of nodes as well as its external controller. And edge set E_T contains three types of edges:

1. The edges from node i_t to node i_{t+1} during the neighbouring time stamps.
2. The edges from node i_t to node j_{t+1} at time stamp t .
3. The edges from the external controller I_t^o to node o_{t+1} at time stamp t , where $o \in V_{\mathcal{G}}$ denotes the directly connected node to controller I^o .

Figure 13.6 gives an example of temporal network and its corresponding time-ordered graph. The temporal network has 4 nodes $\{A, B, C, D\}$ and an external controller I^o . Figure 13.6a illustrates the topology of the temporal graph from $t = 1$ to 4. The numbers in parentheses of Fig. 13.6a represent when the edges are valid. In Fig. 13.6b, each node (including the controller) in Fig. 13.6a has $4 + 1 = 5$ duplications (e.g. A in Fig. 13.6a while A_1, A_2, A_3, A_4, A_5 in Fig. 13.6b). The dashed edges in Fig. 13.6b are from a node to itself during the neighbouring time stamps. The black (dark) edges are the edges between nodes and controller I^o . The blue (light) edges are the edges between different nodes. Weights of interactions (the blue ones) are labeled by $a_{11}, a_{12}, \dots, d_{41}, d_{42}$ in Fig. 13.6b, and without loss of generality, we denote the weights of other edges (the dashed ones and black ones) as “1”. Therefore, by enlarging the scale of the network via duplication of nodes at different time slots, temporal information remains.

Definition 13.9 Given the controller subset $S_1 = \{I_0^o, I_1^o, \dots, I_T^o\}$ and node $i_{T+1} \in S_2 = \{i_{T+1}, 2_{T+1}, \dots, |V_{\mathcal{G}}|_{T+1}\}$ of $N(\mathcal{G}, T)$, which correspond to controller I^o and node $i \in V_{\mathcal{G}}$ of \mathcal{G} , respectively. If, in $N(\mathcal{G}, T)$, there exists a path from $I_t^o \in S_1$ to $i_{T+1} \in S_2$, node i of \mathcal{G} is reachable from controller I^o at time t , and the set of such reachable nodes in $V_{\mathcal{G}}$ is the reachable subset of controller I^o of \mathcal{G} .

The dynamic communicability matrix [16] to quantify the effectiveness of a node to broadcast and receive messages in a network is defined as

$$Q = (I + aA_1)(I + aA_2) \dots (I + aA_T) \quad (13.12)$$

where matrix A_t , $1 \leq t \leq T$ is the adjacency matrix of the graph at time stamp t , and $0 < a < 1/\rho$, ρ is the maximum spectral radius of matrices A_t . Similarly, we define the temporal communicability matrix to quantify the reachability of a temporal network’s single controller at different time stamps as

$$Q_t = (I^* + a_t A_t^*)(I^* + a_{t+1} A_{t+1}^*) \dots (I^* + a_T A_T^*) \quad (13.13)$$

where $A_t^* = \begin{pmatrix} 0 & (b^{(o)})' \\ \mathbf{0}_{N \times 1} & A_t \end{pmatrix}$ is the adjacency matrix of the temporal network at time stamp t with single controller I^o , which is placed on node o , and $I^* = \begin{pmatrix} 0 & \mathbf{0}_{1 \times N} \\ \mathbf{0}_{N \times 1} & I_{N \times N} \end{pmatrix}$. $\{Q_t\}_{i,j}$ represents the reachability from node i to node j . The reachability of controller I^o at time stamp t is presented by the o th row of matrix Q_t , i.e., $\{Q_t\}_{o,\forall}$. Collect row $\{Q_t\}_{o,\forall}$, $t = 1, 2, \dots, T$, as matrix $W^* = [(\{Q_1\}_{o,\forall})', (\{Q_2\}_{o,\forall})', \dots, (\{Q_T\}_{o,\forall})']$. Therefore, W^* represents the reachability of the controller subset S_1 .

Theorem 13.3 *The reachability of the controller subset S_1 is equivalent to the reachability of the controller of \mathcal{G} , i.e.,*

$$W^* = [(\{Q_1\}_{o,\forall})', (\{Q_2\}_{o,\forall})', \dots, (\{Q_T\}_{o,\forall})'] = \begin{pmatrix} \mathbf{0}_{1 \times T} \\ W_c \end{pmatrix} \quad (13.14)$$

Proof Denote the partitioned matrix $A_{N(\mathcal{G},T)} = \begin{pmatrix} B_{(1,1)} & B_{(1,2)} & \cdots & B_{(1,N)} \\ B_{(2,1)} & B_{(2,2)} & \cdots & B_{(2,N)} \\ \vdots & \vdots & \ddots & \vdots \\ B_{(N,1)} & B_{(N,2)} & \cdots & B_{(N,N)} \end{pmatrix}$ the adjacency matrix of $N(\mathcal{G}, T)$. For each block $B_{(i,j)}$ (size $(T+1) \times (T+1)$) in matrix $A_{N(\mathcal{G},T)}$, if there's a directed edge starting from node i_t to node j_{t+1} in $N(\mathcal{G}, T)$, where $t=1, 2, \dots, T$, we have $\{B_{(i,j)}\}_{t,t+1} \neq 0$ and $\{A_{N(\mathcal{G},T)}\}_{i(T+1)+t, j(T+1)+t+1} \neq 0$. A non-zero element at row i column j (denoted as (i, j)) of the product of matrices $(A)^k$ is the reachability from node i to node j if $\{(A)_k\}_{i,j} \neq 0$. Besides, the lengths of paths in graph $N(\mathcal{G}, T)$ are no more than $T+1$. Therefore, the reachability of each controller I_t^o in $S_1 = \{I_0^o, I_1^o, \dots, I_T^o\}$ is the row $(t+1)$ of $(T+1-t)$ th power of $A_{N(\mathcal{G},T)}$, i.e., $\{(A_{N(\mathcal{G},T)})^{T+1-t}\}_{t+1,\forall}$, $t = 0, 1, \dots, T$. For each column of matrix W_c , we have $G_T \dots G_2 H_1 = [(G_T \dots G_2)']' H_1 = [G_2' \dots G_T']' H_1 = [(I + A_2) \dots (I + A_T)]' H_1$. Recall the definition of matrix Q_t , we know that $\{Q_t\}_{i,j}$ describes the reachability from node i to node j . Therefore, the reachability of controller I_t^o at time t is equivalent to the controlled row, i.e. the o th row of matrix Q_t (denoted as $\{Q_t\}_{o,\forall}$). We can rewrite matrix W_c in the form of reachability as:

$$W^* = [(\{Q_1\}_{o,\forall})', (\{Q_2\}_{o,\forall})', \dots, (\{Q_T\}_{o,\forall})'] = \begin{pmatrix} \mathbf{0}_{1 \times T} \\ W_c \end{pmatrix}. \quad \square$$

By Theorem 13.3, we easily get $\text{rank}(W^*) = \text{rank}(W_c)$.

For example, in Fig. 13.6, $\{Q_1\}_{1,\forall} = [0, 1+a_{21}c_{31}b_{42}, a_{21}, a_{21}c_{31}+b_{41}, a_{21}d_{41}]$, $\{Q_2\}_{1,\forall} = [0, 1, 0, b_{41}, 0]$, $\{Q_3\}_{1,\forall} = [0, 1, 0, b_{41}, 0]$, and $\{Q_4\}_{1,\forall} = [0, 1, 0, 0, 0]$. By Theorem 13.3, $W^* = [(\{Q_1\}_{1,\forall})', (\{Q_2\}_{1,\forall})', (\{Q_3\}_{1,\forall})', (\{Q_4\}_{1,\forall})']$.

$$(\{Q_4\}_{1,\forall})' = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1+a_{21}c_{31}b_{42} & 1 & 1 & 1 \\ a_{21} & 0 & 0 & 0 \\ a_{21}c_{31}+b_{41} & b_{41} & b_{41} & 0 \\ a_{21}d_{41} & 0 & 0 & 0 \end{pmatrix}.$$

13.5.3 Structured Temporal Tree

By the Breadth-First Search (BFS) algorithm, we get the spanning trees (denoted as ST_t) of the time-ordered graph $N(\mathcal{G}, T)$, which is rooted at node I_t^o , $t = 1, 2, \dots, T$.

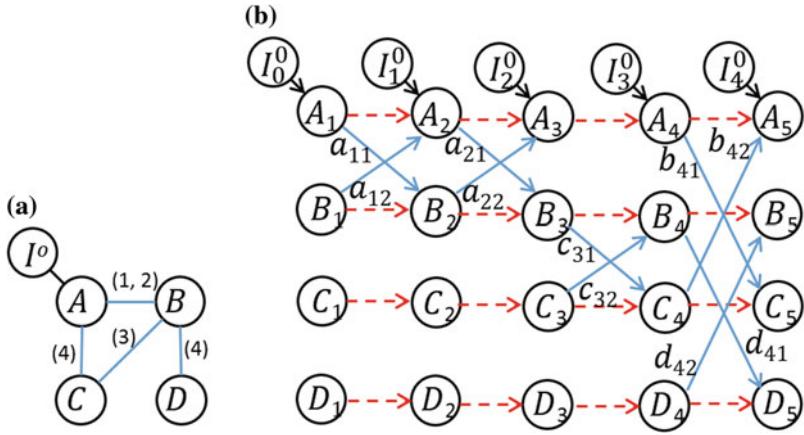


Fig. 13.6 [50] **a** Temporal Network with a single controller located on node \$A\$, **b** the corresponding Time-Ordered Graph (TOG). The dashed, black (dark) and blue (light) edges stand for the flows of time order, the connection with the single controller and the interactions of individuals, respectively. Weights of interactions (the blue ones) are labeled by characters \$a_{11}, a_{12}, \dots, d_{41}, d_{42}\$ in **(b)**, and without loss of generality, we denote the weight of other edges (the dashed ones and black ones) as ‘1’

Definition 13.10 A temporal tree \$TT_t, t = 1, 2, \dots, T\$ is such a tree of the temporal network which gathers redundant node duplications corresponding to the spanning tree \$ST_t\$ of the time-ordered graph.

Figure 13.7 illustrates the temporal trees and the BFS spanning trees of the temporal network in Fig. 13.2. Since the temporal network in Fig. 13.2 has 4 time stamps, we get 4 BFS spanning trees rooted at \$I_t, t = 1, 2, 3, 4\$, from the corresponding time-ordered graph, respectively. After gathering node duplications (e.g., from nodes \$A_2, A_3, A_4, A_5\$ in (b) to node \$A\$ in Fig. 13.2a), we get 4 temporal trees finally.

Definition 13.11 The reachability vector of temporal tree \$TT_t\$ is denoted as \$R_{TT_t}\$, \$R_{TT_t} \in R^{(V_G+1) \times 1}, t = 1, 2, \dots, T\$. Since every temporal tree is rooted at controller \$I^0\$ which is placed on node \$o\$, the first row in \$R_{TT_t}\$ is zero, and the row \$o+1\$ is 1, and rest rows \$i+1, i \neq o, 1 \leq i \leq |V_G|\$ are equal to the product of edges' weights starting from node \$o\$ to node \$i\$, or zero, if node \$i\$ is not on the temporal tree \$TT_t\$. Matrix \$W^R = [R_{TT_1} \ R_{TT_2} \ \dots \ R_{TT_T}] \in R^{(V_G+1) \times T}\$ records all the temporal trees from time stamps 1 to \$T\$.

Theorem 13.4 \$rank(W^R) = rank(W^*)\$.

Proof With Theorem 13.3 and Definition 13.10, each temporal tree \$TT_t\$ is associated with the BFS spanning tree \$ST_t\$ in TOG. Since each \$ST_t\$ is a leading tree compared with \$ST_{t+1}\$ (refer to the definition of BFS spanning tree with the TOG model), each temporal tree \$TT_t\$ is a leading tree compared with \$TT_{t+1}\$.

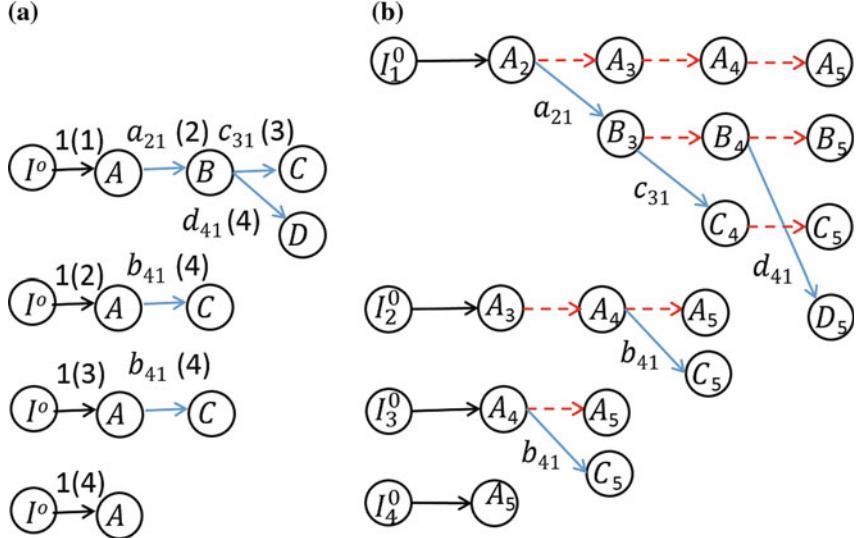


Fig. 13.7 [50] **a** The temporal trees of Fig. 13.6 **a** at time stamps 1, 2, 3 and 4. **b** The BFS spanning trees of TOG. The dashed, black (dark) and blue (light) edges stand for the flows of time order, the connection with the single controller and the interactions of individuals, respectively. The numbers with parenthesis in **(a)** denote time stamps when these edges are valid. Weights of interactions (the blue ones) are labeled by characters $a_{11}, a_{12}, \dots, d_{41}, d_{42}$ in **(a)** and **(b)**, and without loss of generality, we denote the weight of other edges (the dashed ones and black ones) as ‘1’

There are two cases to yield a leading temporal tree : (i) adding more nodes in TT_t , i.e., $|V_{TT_t}| > |V_{TT_{t+1}}|$, (ii) adding new paths among the existing nodes, i.e., $|V_{TT_t}| = |V_{TT_{t+1}}|$. In case (i), if there’s only one temporal tree in the temporal network, it is obvious that $\text{rank}(W^R) = \text{rank}(R_{TT}) = \text{rank}(W^*) = 1$. Suppose that the number of temporal trees is n , and $\text{rank}(W^*) = \text{rank}(W^R) = n$. When the number of temporal trees is $n+1$, we have $\text{rank}(W^R) = \text{rank} \begin{pmatrix} \mathbf{0}_{1 \times n} & 0 \\ W_{n \times n}^* & (\ddagger)_{n \times 1} \\ \mathbf{0}_{(|V_G|-n) \times n} & (\ddagger)_{(|V_G|-n) \times 1} \end{pmatrix} = \text{rank}(W^*) = n+1$, where (\ddagger) denotes a nonzero vector.

In case (ii), each unique edge in leading tree TT_t , which doesn’t appear in the following temporal tree TT_{t+1} , implies unique paths to the existing nodes on TT_t . Therefore, “removing” those “old” edges has no impact on the rank of matrix W^* , which means that $\text{rank}(W^R) = \text{rank}(P_1 W^R T_1) = \text{rank}(W^*) = \text{rank}(P_2 W^* T_2)$, where P_1, P_2, T_1 and T_2 are properly selected linear transformation matrices. \square

Corollary 13.1 $\text{rank}(W^R) = \text{rank}(W_c)$

Proof By Theorems 13.3 and 13.4, we directly get $\text{rank}(W^R) = \text{rank}(W^*) = \text{rank}(W_c)$. \square

For the example in Fig. 13.7, the external controller is placed on node A, so

$$W^R = [R_{TT_1}, R_{TT_2}, R_{TT_3}, R_{TT_4}] = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ a_{21} & 0 & 0 & 0 \\ a_{21}c_{31} & b_{41} & b_{41} & 0 \\ a_{21}d_{41} & 0 & 0 & 0 \end{pmatrix}.$$

With proper linear transformation matrices T_1, T_2, P_1, P_2 ,

$$P_1 W^R T_1 = P_1 \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ a_{21} & 0 & 0 & 0 \\ a_{21}c_{31} & b_{41} & b_{41} & 0 \\ a_{21}d_{41} & 0 & 0 & 0 \end{pmatrix} T_1 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ a_{21} & 0 & 0 & 0 \\ a_{21}c_{31} & 0 & b_{41} & 0 \\ a_{21}d_{41} & 0 & 0 & 0 \end{pmatrix}.$$

$$\text{On the other hand, } P_2 W^* T_2 = P_2 \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 + a_{21}c_{31}b_{42} & 1 & 1 & 1 \\ a_{21} & 0 & 0 & 0 \\ a_{21}c_{31} + b_{41} & b_{41} & b_{41} & 0 \\ a_{21}d_{41} & 0 & 0 & 0 \end{pmatrix} T_2 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ a_{21} & 0 & 0 & 0 \\ a_{21}c_{31} & 0 & b_{41} & 0 \\ a_{21}d_{41} & 0 & 0 & 0 \end{pmatrix}.$$

It is obvious that $\text{rank}(W^R) = \text{rank}(P_1 W^R T_1) = \text{rank}(W^*) = \text{rank}(P_2 W^* T_2)$.

Definition 13.12 Homogeneously structured trees are the temporal trees with the same structure (the locations of non-zero elements in the adjacency matrix are the same). Otherwise, the temporal trees are heterogeneously structured trees.

According to Definition 13.12, W^R can be split into two parts as

$$W^R = (W^D \ W^S) = \begin{pmatrix} \mathbf{0}_{1 \times T^D} & \mathbf{0}_{1 \times T^S} \\ W_c^D & W_c^S \end{pmatrix} \quad (13.15)$$

Matrix $W^D = (\mathbf{0}_{T^D \times 1} \ (W_c^D)')$ of size $(|V_G|+1) \times T^D$ represents heterogeneously structured trees in matrix W^R , and T^D denotes the number of heterogeneously structured trees. Matrix $W^S = (\mathbf{0}_{T^S \times 1} \ (W_c^S)')$ of size $(|V_G|+1) \times T^S$ represents homogeneously structured trees, and T^S denotes the number of homogeneously structured trees. Besides, $T^D + T^S = T$.

13.5.4 Controllability Bounds of Categories of Structured Temporal Trees and Networks

In the remaining part of this chapter, we further categorize the structured temporal trees to estimate the bounds of structural controllability of a temporal network.

13.5.4.1 Homogeneously Structured Trees

Definition 13.13 Define the homogeneously structured trees with independent adjacency matrices as independent trees, where the value of each non-zero element can be taken freely. Other homogeneously structured trees without independent adjacency matrices are defined as interdependent trees.

Figure 13.8 gives some examples of homogeneously structured trees. Triple (i, j, t) denotes the edge from node i to node j at time stamp t . Temporal trees in Fig. 13.8a are independent trees because the weights of edges among nodes are independent on each other. However, since the weights of edges $(B, C, 5)$, $(B, D, 5)$ and $(B, E, 5)$ in Fig. 13.8b, and edges $(B, C, 5)$, $(B, D, 5)$ in Fig. 13.8c are the same, the temporal trees in Fig. 13.8b, c are interdependent trees.

According to Definition 13.13,

$$W^S = (W_1^S \ W_2^S \ \dots \ W_q^S) \quad (13.16)$$

In matrix W^S , each W_m^S , $m = 1, 2, \dots, q$, represents a collection of homogeneously structured trees. Denote $|V_{W_m^S}|$ the number of nodes in matrix W_m^S . We have $V_{W_m^S} \neq V_{W_{m'}^S}$ for $\forall m \neq m'$). Therefore, W_m^S can be written as:

$$W_m^S = \left(\begin{array}{c|c} \mathbf{0}_{1 \times T_{S_m,1}^S} \cdots \mathbf{0}_{1 \times T_{S_m,p(m)}^S} & \mathbf{0}_{1 \times T_{S_m,p(m)+1}^S} \\ \hline W_{S_m,1}^S \cdots W_{S_m,p(m)}^S & W_{S_m,p(m)+1}^S \end{array} \right) = (W_{S_m,1}^{S*} \cdots W_{S_m,p(m)}^{S*} | W_{S_m,p(m)+1}^{S*}) \quad (13.17)$$

In W_m^S , each $W_{S_m,w}^{S*} = (\mathbf{0}_{T_{S_m,w}^S \times 1} \ (W_{S_m,w}^S)')$, $w = 1, 2, \dots, p(m)$, of size $(|V_G| + 1) \times T_{S_m,w}^S$ is a collection of interdependent trees with interdependent edges, and $I_{m,w} \neq I_{m,w'}$ for $\forall w \neq w'$, where $I_{m,w}$ represents the collection of interdepen-

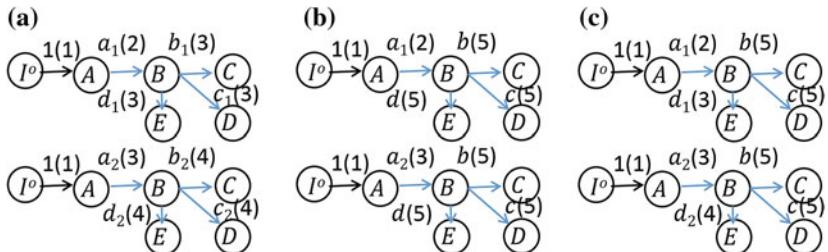


Fig. 13.8 [50] Examples of homogeneously structured trees. **a** Independent trees, **b** interdependent trees with three interdependent edges $(B, C, 5)$, $(B, D, 5)$ and $(B, E, 5)$, and **c** interdependent trees with two interdependent edges $(B, C, 5)$ and $(B, D, 5)$. Triple (i, j, t) denotes that the edge from node i to node j at time stamp t . Parameters $a_1, a_2, b, b_1, b_2, c, c_1, c_2, d, d_1$ and d_2 denote the weights of edges, and numbers in parentheses denote active time stamps of edges

dent edges in matrix $W_{S_m,w}^{S*}$. $W_{S_m,p(m)+1}^{S*} = \left(\mathbf{0}_{T_{S_m,p(m)+1}^S \times 1} (W_{S_m,p(m)+1}^S)' \right)'$ of size $(|V_{\mathcal{G}}| + 1) \times T_{S_m,p(m)+1}^S$ represents the part of independent trees. Therefore, $\sum_{m=1}^q \sum_{w=1}^{p(m)+1} T_{S_m,w}^S = T^S$, and $|V_{W_m^S}| = |V_{W_{S_m,w}^{S*}}|$, $w = 1, 2, \dots, p(m) + 1$, where $T_{S_m,w}^S$ is the number of trees in a collection of homogeneously structured trees with interdependent edges, and $|V_{W_m^S}|$ and $|V_{W_{S_m,w}^{S*}}|$ are the number of nodes in matrices W_m^S and $W_{S_m,w}^{S*}$, respectively.

Theorem 13.5 Denote a collection of independent trees by $W_{S_m,p(m)+1}^{S*}$, we have $\text{rank}(W_{S_m,p(m)+1}^{S*}) = \min(|V_{W_m^S}|, T_{S_m,p(m)+1}^S)$, where $|V_{W_m^S}|$ denotes the number of nodes in matrix $W_{S_m,p(m)+1}^{S*}$.

Proof The non-zero elements in the reachability vectors of independent trees from the controller I^o , i.e. $[R_{TT_1}^S, R_{TT_2}^S, \dots]$, are all independent. Therefore, there is always a square sub-matrix in $W_{S_m,p(m)+1}^{S*}$ with the size of $\min(|V_{W_m^S}|, T_{S_m,p(m)+1}^S) \times \min(|V_{W_m^S}|, T_{S_m,p(m)+1}^S)$, whose elements are both non-zero and independent. Therefore, $\text{rank}(W_{S_m,p(m)+1}^{S*}) = \min(|V_{W_m^S}|, T_{S_m,p(m)+1}^S)$. \square

For example, in Fig. 13.8a, matrix $W_{S_m,p(m)+1}^{S*}$ representing independent trees is $W_{S_m,p(m)+1}^{S*} = \begin{pmatrix} 0 & 1 & a_1 & a_1 b_1 & a_1 c_1 & a_1 d_1 \\ 0 & 1 & a_2 & a_2 b_2 & a_2 c_2 & a_2 d_2 \end{pmatrix}$, whose rank is 2. So $\text{rank}(W_{S_m,p(m)+1}^{S*}) = T_{S_m,p(m)+1}^S = 2 < |V_{W_m^S}| = 5$. In general cases, if $T_{S_m,p(m)+1}^S = n > |V_{W_m^S}| = 5$, matrix $W_{S_m,p(m)+1}^{S*} = \begin{pmatrix} 0 & 0 & \cdots & 0 & 0 \\ 1 & 1 & \cdots & 1 & 1 \\ a_1 & a_2 & \cdots & a_n & a_n \\ a_1 b_1 & a_2 b_2 & \cdots & a_{n-1} b_{n-1} & a_n b_n \\ a_1 c_1 & a_2 c_2 & \cdots & a_{n-1} c_{n-1} & a_n c_n \\ a_1 d_1 & a_2 d_2 & \cdots & a_{n-1} d_{n-1} & a_n d_n \end{pmatrix}$, and $\text{rank}(W_{S_m,p(m)+1}^{S*}) = |V_{W_m^S}| = 5 < T_{S_m,p(m)+1}^S = n$.

Theorem 13.6 Denote a collection of interdependent trees by $W_{S_m,w}^{S*}$, we have $\text{rank}(W_{S_m,w}^{S*}) = \min(|V_{W_m^S}| - |I_{m,w}|, T_{S_m,w}^S)$, ($w = 1, 2, \dots, p(m)$), where $|V_{W_m^S}|$ denotes the number of nodes, and $|I_{m,w}|$ denotes the number of interdependent edges in $W_{S_m,w}^{S*}$.

Proof For simplicity without loss of generality, we first prove the case of two trees as illustrated in Fig. 13.8b. Here $|I_{m,w}| = 3$, i.e. interdependent edges $(B, C, 5)$, $(B, D, 5)$ and $(B, F, 5)$. The rank of corresponding matrix $\text{rank}(W_{S_m,w}^{S*}) = \text{rank} \begin{pmatrix} 0 & 1 & a_1 & a_1 b & a_1 c & a_1 d \\ 0 & 1 & a_2 & a_2 b & a_2 c & a_2 d \end{pmatrix}' = \text{rank} \begin{pmatrix} 0 & 1 & a_1 & a_1 b & a_1 c & a_1 d \\ 0 & (a_1 - a_2)/a_1 & 0 & 0 & 0 & 0 \end{pmatrix}$. The dependence of elements in matrix is caused by the interdependent edges of trees. Therefore, $\text{rank}(W_{S_m,w}^{S*}) = 2 = |V_{W_m^S}| - |I_{m,w}| = T_{S_m,w}^S$.

More generally, when extending to the case of n trees,

$$\begin{aligned} \text{rank}(W_{S_m,w}^{S*}) &= \text{rank} \begin{pmatrix} 0 & 0 & \cdots & 0 & 0 \\ 1 & 1 & \cdots & 1 & 1 \\ a_1 & a_2 & \cdots & a_{n-1} & a_n \\ a_1b & a_2b & \cdots & a_{n-1}b & a_nb \\ a_1c & a_2c & \cdots & a_{n-1}c & a_nc \\ a_1d & a_2d & \cdots & a_{n-1}d & a_nd \end{pmatrix} \\ &= \text{rank} \begin{pmatrix} 0 & 0 & \cdots & 0 & 0 \\ 1 & (a_1 - a_2)/a_1 & \cdots & (a_1 - a_{n-1})/a_1 & (a_1 - a_n)/a_1 \\ a_1 & 0 & \cdots & 0 & 0 \\ a_1b & 0 & \cdots & 0 & 0 \\ a_1c & 0 & \cdots & 0 & 0 \\ a_1d & 0 & \cdots & 0 & 0 \end{pmatrix}, \end{aligned}$$

and $\text{rank}(W_{S_m,w}^{S*}) = 2 = |V_{W_m^S}| - |I_{m,w}| < T_{S_m,w}^S = n$.

Similarly, for the trees in Fig. 13.8c, $\text{rank}(W_{S_m,w}^{S*}) = \begin{pmatrix} 0 & 1 & a_1 & a_1b & a_1c & a_1d_1 \\ 0 & 1 & a_2 & a_2b & a_2c & a_2d_2 \end{pmatrix}' = \text{rank} \begin{pmatrix} 0 & 1 & a_1 & a_1b & a_1c & a_1d_1 \\ 0 & (a_1 - a_2)/a_1 & 0 & 0 & 0 & (d_2 - d_1)a_2 \end{pmatrix}',$ and $\text{rank}(W_{S_m,w}^{S*}) = 2 = T_{S_m,w}^S < |V_{W_m^S}| - |I_{m,w}| = 3$. When extending to the case of n trees,

$$\begin{aligned} \text{rank}(W_{S_m,w}^{S*}) &= \text{rank} \begin{pmatrix} 0 & 0 & \cdots & 0 & 0 \\ 1 & 1 & \cdots & 1 & 1 \\ a_1 & a_2 & \cdots & a_{n-1} & a_n \\ a_1b & a_2b & \cdots & a_{n-1}b & a_nb \\ a_1c & a_2c & \cdots & a_{n-1}c & a_nc \\ a_1d_1 & a_2d_2 & \cdots & a_{n-1}d_{n-1} & a_nd_n \end{pmatrix} \\ &= \text{rank} \begin{pmatrix} 0 & 0 & \cdots & 0 & 0 \\ 1 & (a_1 - a_2)/a_1 & \cdots & (a_1 - a_{n-1})/a_1 & (a_1 - a_n)/a_1 \\ a_1 & 0 & \cdots & 0 & 0 \\ a_1b & 0 & \cdots & 0 & 0 \\ a_1c & 0 & \cdots & 0 & 0 \\ a_1d_1 & (d_2 - d_1)a_2 & \cdots & (d_{n-1} - d_1)a_{n-1} & (d_n - d_1)a_n \end{pmatrix}, \end{aligned}$$

and $\text{rank}(W_{S_m,w}^{S*}) = 3 = |V_{W_m^S}| - |I_{m,w}| < T_{S_m,w}^S = n$. \square

Theorem 13.7 Denote homogeneously structured trees by $W_{S_m,w}^{S*}$, $w = 1, 2, \dots, p(m) + 1$,

$$\begin{aligned} \text{rank}(W_m^S) &= \min \{ \min [\sum_{w=1}^{p(m)} (\text{rank}(W_{S_m,w}^{S*}))], \max_{w=1}^{p(m)} \{|V_{W_m^S}| - |I_{m,w}|\}] + \\ &\quad \text{rank}(W_{S_m,p(m)+1}^{S*}), |V_{W_m^S}| \} \end{aligned} \tag{13.18}$$

where $|V_{W_m^S}|$ denotes the number of nodes, and $|I_{m,w}|$ denotes the number of interdependent edges in $W_{S_m,w}^{S*}$, $w = 1, 2, \dots, p(m)$.

Proof The outsider function $\min\{\}$ sets a limitation that the rank of matrix W_m^S could never be larger than the number of nodes which is also equal to the number of rows in matrix W_m^S .

In $\min[\sum_{w=1}^{p(m)}(\text{rank}(W_{S_m,w}^{S*}))]$, $\max_{w=1}^{p(m)}\{|V_{W_m^S}| - |I_{m,w}|\}] + \text{rank}(W_{S_m,p(m)+1}^{S*})$, since independent trees ($W_{S_m,p(m)+1}^{S*}$) always contribute to the rank of W_m^S , there is an added $\text{rank}(W_{S_m,p(m)+1}^{S*})$ when calculating the rank.

$\min[\sum_{w=1}^{p(m)}(\text{rank}(W_{S_m,w}^{S*}))]$, $\max_{w=1}^{p(m)}\{|V_{W_m^S}| - |I_{m,w}|\}]$ represents the part of interdependent trees in W_m^S . Since the trees from different collections $W_{S_m,w}^{S*}$ have independent structures, when $\sum_{w=1}^{p(m)}(\text{rank}(W_{S_m,w}^{S*})) \leq \max_{w=1}^{p(m)}(|V_{W_m^S}| - |I_{m,w}|)$, the rank of the interdependent trees' part in W_m^S is equal to $\sum_{w=1}^{p(m)}(\text{rank}(W_{S_m,w}^{S*}))$. Otherwise, if $\sum_{w=1}^{p(m)}(\text{rank}(W_{S_m,w}^{S*})) > \max_{w=1}^{p(m)}(|V_{W_m^S}| - |I_{m,w}|)$, the rank of the interdependent trees' part in W_m^S is less than or equal to the maximum rank among every collection. \square

$$\text{For example, in Fig. 13.8b, c, } \text{rank}\left(W_{S_m,1}^S W_{S_m,2}^S\right) = \text{rank} \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ a_1 & a_2 & a_1 & a_2 \\ a_1 b & a_2 b & a_1 b & a_2 b \\ a_1 c & a_2 c & a_1 c & a_2 c \\ a_1 d & a_2 d & a_1 d_1 & a_2 d_2 \end{pmatrix}$$

$$= \text{rank} \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & (a_1 - a_2)/a_1 & 0 & (a_1 - a_2)/a_1 \\ a_1 & 0 & 0 & 0 \\ a_1 b & 0 & 0 & 0 \\ a_1 c & 0 & 0 & 0 \\ a_1 d & 0 & a_1(d_1 - d) & a_2(d_2 - d) \end{pmatrix}. \text{ So } \text{rank}(W_{S_m,1}^S W_{S_m,2}^S) = 3 \\ \text{rank}(W_{S_m,1}^S) + \text{rank}(W_{S_m,2}^S) = 4.$$

Lemma 13.1 Denote the collections of homogeneously structured trees and the maximum structurally controllable subspace of homogeneously structured trees by matrices W_m^S , $m = 1, 2, \dots, q$, and $S_{M(o)}^S$ respectively,

$$\max_{m=1}^q \text{rank}(W_m^S) \leq S_{M(o)}^S = \text{rank}(W^S) \leq \sum_{m=1}^q \text{rank}(W_m^S) \quad (13.19)$$

Proof By Eq. (13.16) and Theorem 13.7, we can easily get Lemma 13.1. \square

13.5.4.2 Heterogeneously Structured Trees

Definition 13.14 Define the heterogeneous structured trees having the same nodes as heterogeneous structured trees with the same nodes. Other heterogeneous structured trees are heterogeneous structured trees with different nodes.

According to Definition 13.14,

$$W^D = \left(\begin{array}{c|c} \mathbf{0}_{1 \times T_{S_1}^D} & \mathbf{0}_{1 \times T_{S_2}^D} \cdots \mathbf{0}_{1 \times T_{S_k}^D} \\ \hline W_{S_1}^D & W_{S_2}^D \cdots W_{S_k}^D \\ & W_{S_{k+1}}^D \end{array} \right) = \left(W_{S_1}^{D*} \ W_{S_2}^{D*} \cdots W_{S_k}^{D*} \mid W_{S_{k+1}}^{D*} \right) \quad (13.20)$$

In matrix W^D , each $W_{S_l}^{D*} = (\mathbf{0}_{T_{S_l}^D \times 1} \ (W_{S_l}^D)')'$, $l = 1, 2, \dots, k$, of size $(|V_{\mathcal{G}}| + 1) \times T_{S_l}^D$ is a collection of heterogeneous trees with the same nodes ($V_{W_{S_l}^{D*}} \neq V_{W_{S_{l'}}^{D*}}$

for $\forall l \neq l'$), and $W_{S_{k+1}}^{D*} = (\mathbf{0}_{T_{S_{k+1}}^D \times 1} \ (W_{S_{k+1}}^D)')' = [R_{TT_1}^D, R_{TT_2}^D, \dots, R_{TT_{T_{S_{k+1}}^D}}^D]$ of size $(|V_{\mathcal{G}}| + 1) \times T_{S_{k+1}}^D$ represents the heterogeneous trees with different nodes. $\sum_{l=1}^{k+1} T_{S_l}^D = T^D$.

Theorem 13.8 Denote a collection of heterogeneous trees with the same nodes by matrix $W_{S_l}^{D*}$, $\text{rank}(W_{S_l}^{D*}) = \min(|V_{W_{S_l}^{D*}}|, T_{S_l}^D)$, $l = 1, 2, \dots, k$, where $|V_{W_{S_l}^{D*}}|$ is the number of nodes in matrix $W_{S_l}^{D*}$.

Proof Note that the heterogeneous trees have different structures from each other. Therefore, each heterogeneous tree with the same nodes implies that it has unique edges that other trees don't have, i.e., these trees have the same reachability with different paths to reach the same node. When $T_{S_l}^D = 1$, we have $\text{rank}(W_{S_l}^{D*[1]}) = 1$. Otherwise, when $T_{S_l}^D = n \leq |V_{W_{S_l}^{D*}}|$, due to the unique edges which presents the independent parameters, $\text{rank}(W_{S_l}^{D*}) = T_{S_l}^D = n$. Similarly, when $|V_{W_{S_l}^{D*}}| \leq T_{S_l}^D = n$, $\text{rank}(W_{S_l}^{D*})$ is less than or equal to $|V_{W_{S_l}^{D*}}|$. Therefore, we have $\text{rank}(W_{S_l}^{D*}) = \min(|V_{W_{S_l}^{D*}}|, T_{S_l}^D)$. \square

Theorem 13.9 Denote the heterogeneous trees with different nodes by matrix $W_{S_{k+1}}^{D*}$, $\text{rank}(W_{S_{k+1}}^{D*}) = T_{S_{k+1}}^D$.

Proof When $T_{S_{k+1}}^D = 1$, obviously $\text{rank}(W_{S_{k+1}}^{D*[1]}) = 1$. Suppose that when $T_{S_{k+1}}^D = n$, $\text{rank}(W_{S_{k+1}}^{D*[n]}) = n$. When $T_{S_{k+1}}^D = n + 1$, a heterogeneous tree with different nodes is added into matrix $W_{S_{k+1}}^{D*[n]}$, and get matrix $W_{S_{k+1}}^{D*[n+1]}$. Since the added tree has at least one unique node that other trees don't have, there is always at least one new nonzero part at column $n + 1$ and row r ($n < r \leq |V_{\mathcal{G}}| + 1$) in matrix $W_{S_{k+1}}^{D*[n+1]}$. So $\text{rank}(W_{S_{k+1}}^{D*[n+1]}) = \text{rank}([W_{S_{k+1}}^{D*[n]}, R_{TT}^D]) =$

$$\text{rank} \begin{pmatrix} \mathbf{0}_{1 \times n} & 0 \\ W_{n \times n}^* & (\ddot{\oplus})_{n \times 1} \\ \mathbf{0}_{(|V_G|-n) \times n} & (\ddot{\oplus})_{(|V_G|-n) \times 1} \end{pmatrix} = n + 1, \text{ where } (\ddot{\oplus}) \text{ denotes a non-zero vector. } \square$$

Theorem 13.10 Denote heterogeneously structured trees and the maximum structurally controllable subspace of heterogeneously structured trees by matrices $W_{S_l}^{D*}$, $l = 1, 2, \dots, k+1$, and $S_{M(o)}^D$, respectively, we have

$$\max_{l=1}^{k+1} \{\text{rank}(W_{S_l}^{D*})\} \leq S_{M(o)}^D = \text{rank}(W^D) \leq \sum_{l=1}^{k+1} \text{rank}(W_{S_l}^{D*}) \quad (13.21)$$

Proof Consider the left part, i.e., $\max_{l=1}^{k+1} \{\text{rank}(W_{S_l}^{D*})\} \leq S_{M(o)}^D = \text{rank}(W^D)$. Compared with the trees $TT_1, TT_2, \dots, TT_{T_{S_{k+1}}^D}$ in matrix $W_{S_{k+1}}^{D*} = [R_{TT_1}^D, R_{TT_2}^D, \dots, R_{TT_{T_{S_{k+1}}^D}}^D]$, those trees in matrices $W_{S_l}^{D*}, l = 1, 2, \dots, k$, have different nodes, i.e., $V_{W_{S_l}^{D*}} \neq V_{TT_1} \neq V_{TT_2} \neq \dots \neq V_{TT_{T_{S_{k+1}}^D}}$, and $V_{W_{S_l}^{D*}} \neq V_{W_{S_l'}^{D*}}$ for $\forall l \neq l'$. We have $\text{rank}(W^D) = \max_{l=1}^{k+1} \{\text{rank}(W_{S_l}^{D*})\}$ when there exists a tree contains all nodes, and it has the maximum rank. Therefore, more generally, we have $\text{rank}(W^D) \geq \max_{l=1}^{k+1} \{\text{rank}(W_{S_l}^{D*})\}$.

For the right part, i.e., $S_{M(o)}^D = \text{rank}(W^D) \leq \sum_{l=1}^{k+1} \text{rank}(W_{S_l}^{D*})$, if the heterogeneous trees have totally different nodes from each other, we have

$$W^D = \begin{pmatrix} \mathbf{0}_{1 \times T_{S_1}^D} & \mathbf{0}_{1 \times T_{S_2}^D} & \cdots & \mathbf{0}_{1 \times T_{S_{k+1}}^D} \\ 1 & 1 & \cdots & 1 \\ W^{*1} & \mathbf{0}_{r_1 \times T_{S_2}^D} & \cdots & \mathbf{0}_{r_1 \times T_{S_{k+1}}^D} \\ \mathbf{0}_{r_2 \times T_{S_1}^D} & W^{*2} & \cdots & \mathbf{0}_{r_2 \times T_{S_{k+1}}^D} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_{r_{k+1} \times T_{S_1}^D} & \mathbf{0}_{r_{k+1} \times T_{S_2}^D} & \cdots & W^{*k+1} \end{pmatrix}, \text{ where row vector } \mathbf{1}_{1 \times T^D} \text{ (row 2 of}$$

matrix W^D) denotes node o where the controller is placed, and matrices $W^{*1}, W^{*2}, \dots, W^{*k+1}$ denote the other nodes of these trees. Since the non-zero elements in $W_{S_l}^{D*}, l = 1, 2, \dots, k+1$, have different locations excluding node o , i.e., $|V_{W_{S_l}^{D*}} \cap V_{W_{S_l'}^{D*}}| = 1$ for $\forall l \neq l'$, each matrix $W_{S_l}^{D*}$ contributes $d = \text{rank}(W_{S_l}^{D*}) = T_{S_l}^D$ to $\text{rank}(W^D)$, and we have $\text{rank}(W^D) = \sum_{l=1}^{k+1} \text{rank}(W_{S_l}^{D*})$. Therefore, more generally, we have $\text{rank}(W^D) \leq \sum_{l=1}^{k+1} \text{rank}(W_{S_l}^{D*})$. \square

After categorizing structured temporal trees with their controllability bounds, we are in the place to estimate the structural controlling centrality of a temporal network as follows, which helps us no need to calculate the rank of W_c directly.

Theorem 13.11 Denote the maximum controlled subspace of heterogeneously structured and homogeneously structured temporal trees by $S_{M(o)}^D$ and $S_{M(o)}^S$ respectively, we have

$$\max(S_{M(o)}^D, S_{M(o)}^S) \leq S_{M(o)} \leq S_{M(o)}^D + S_{M(o)}^S \quad (13.22)$$

Proof By Theorem 13.7, Lemma 13.1 and Theorem 13.10, we easily reach Theorem 13.11. \square

13.5.5 Numerical Verification

We now verify the above bounds in Theorem 13.11 to estimate structural controllability of temporal networks generated from artificial models and empirical data sets.

We first generate a group of temporal networks with the ER random network model [13]. In such an ER temporal network, we randomly connect an edge between a pair of nodes with probability 0.002 for all the $N(N - 1)/2$ (N is the number of nodes in the network) pairs of nodes at a specified time stamp. Repeat this process for 100 rounds at 100 different time stamps, i.e., $t = 1, 2, \dots, 100$. Therefore, we generate 4 groups of ER temporal networks with 40, 60, 80, and 100 nodes correspondingly, and calculate their $\text{rank}(W_c)$ directly. As a comparison, we estimate their upper and lower bounds with Theorem 13.11 as shown in Fig. 13.9, where we clearly observe that the gaps between the upper and lower bounds are minor and cover the directly calculated controlling centrality $\text{rank}(W_c)$ in all 4 groups of ER temporal networks.

We further verify Theorem 13.11 with empirical temporal data sets of “SG-Infectious”, “Fudan WiFi”, which have been introduced in Sect. 13.4, and “HT09”. The empirical temporal data set “HT09” was collected at the ACM Hypertext 2009 conference in Toronto, Canada, which is accessible with the URL of <http://www.sociopatterns.org>. In the generated temporal network of “HT09”, nodes represent the conference attendees, and the temporal edges describe the face-to-face proximities among those attendees.

We generate two empirical temporal networks from “HT09”: one is an empirical temporal network, denoted as “all range”, with all 113 nodes and 9865 edges covering the whole data set “HT09”, and the other one, denoted as “removed”, involves 73 nodes and 3679 edges after removing the nodes having the largest controlling centrality in the temporal network “all range”. Besides, we generate three empirical temporal networks from the data set of “SG-Infectious” by every week. That is to say, temporal network denoted as “Week 1” is generated from the first week of “SG-Infectious”, which has 1321 nodes and 20343 temporal edges activated, temporal network denoted as “Week 2” is generated from the second week of “SG-Infectious”, which has 868 nodes and 13401 temporal edges activated, and temporal

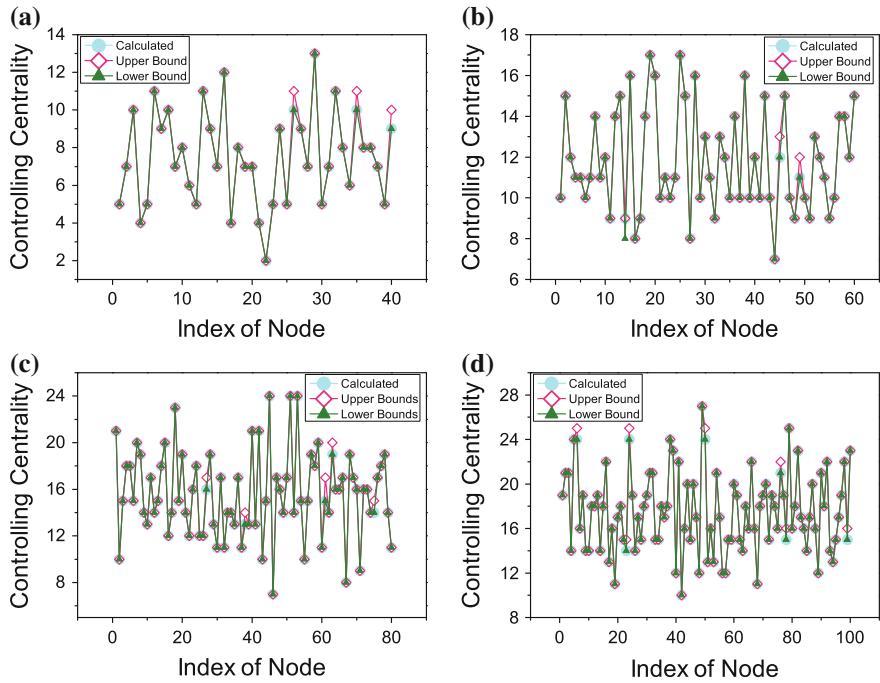


Fig. 13.9 [50] Controlling centrality of synthetic temporal networks with **a** 40 nodes, **b** 60 nodes, **c** 80 nodes, and **d** 100 nodes. For each of the four networks, we randomly generate an edge between a pair of nodes with probability 0.002, and repeat it for all the $N(N - 1)/2$ (N denote the number of nodes in the network) pairs of nodes at a specified time stamp. Repeat this process for 100 rounds at 100 different time stamps, i.e., $t = 1, 2, \dots, 100$. The value of controlling centrality, denoted as ‘Calculated’, is straightly calculated by the computation of matrix W_c in Eq. (13.8), and the upper and lower bounds, denoted as ‘Upper Bound’ and ‘Lower Bound’, respectively, are given by the analytical results in Eq. (13.22)

network “Week 1&2” is generated from the first two weeks from “SG-Infectious”, which has totally 2189 nodes and 33744 temporal edges activated. Similarly, we generate three empirical temporal networks from the data set of “Fudan WiFi”. In more detail, temporal network “Day 1” is generated from the first day of data set “Fudan WiFi”, which has 1120 nodes and 12833 temporal edges, temporal network “Day 2” is generated from the second day of data set “Fudan WiFi”, which has 2250 nodes and 25772 temporal edges, temporal network “713 point” is generated from the records of data set “Fudan WiFi” collected at No. 713 WiFi Access Point, which has 1838 nodes and 27810 temporal edges. All these eight empirical temporal networks although cover different sizes of nodes and edges, their gaps of estimated bounds are relatively very little as shown in Fig. 13.10. Therefore, Theorem 13.11 gives a rather precise estimation of the structural controlling centrality of a temporal network, leaving aside the direct calculation of $\text{rank}(W_c)$.

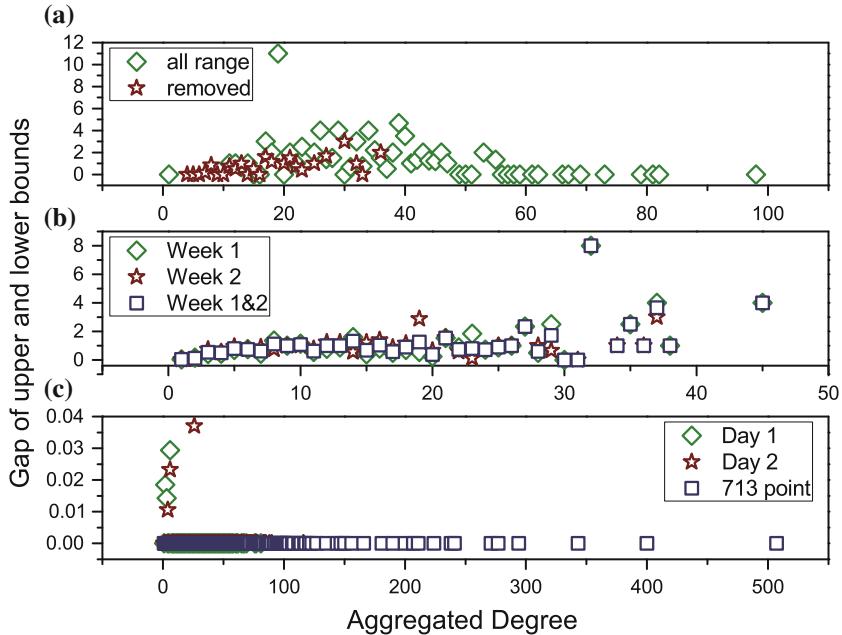


Fig. 13.10 [50] The gap of upper bound and lower bound of controlling centrality. **a** HT09 **b** SG-Infectious **c** Fudan WIFI. For the dataset ‘HT09’, two temporal networks are generated: (i) a temporal network (113 nodes and 9865 edges) with all nodes and edges within record of dataset, denoted as ‘all range’, (ii) a temporal network (73 nodes and 3679 edges) with nodes and edges after removing the most powerful nodes (nodes with the largest controlling centrality) in the temporal network of (i), denoted as ‘removed’. For the dataset ‘SG-Infectious’, three temporal networks are generated: (i) a temporal network (1321 nodes and 20343 edges) with nodes and edges recorded in the first week, denoted as ‘Week 1’, (ii) a temporal network (868 nodes and 13401 edges) with nodes and edges recorded in the second week, denoted as ‘Week 2’, (iii) a temporal network (2189 nodes and 33744 edges) with nodes and edges recorded in the first two weeks, denoted as ‘Week 1&2’. For the dataset ‘Fudan WIFI’, three temporal networks are generated: (i) a temporal network (1120 nodes and 12833 edges) with nodes and edges recorded in the first day, denoted as ‘Day 1’, (ii) a temporal network (2250 nodes and 25772 edges) with nodes and edges recorded in the second day, denoted as ‘Day 2’, (iii) a temporal network (1838 nodes and 27810 edges) with nodes and edges recorded at Access Point No. 713, denoted as ‘713 point’. The upper bound and lower bound of the controlling centrality are given by analytical results in Eq. (13.22), and the gap is given by the absolute value of the difference between the upper bound and lower bound. The aggregated degree of a node is the number of neighbored nodes that the node interacts within the corresponding temporal network. Compared with the sizes of these temporal networks, all the gaps are minor

13.6 Outlook

Understanding the complexity of this networking world is interdependent to the control of complex networks, which, no matter yes or no, is regarded as the ultimate goal to discover, analyze, and predict ubiquitous complex networking systems. No one will deny the significance of the seminar discovery of scale-free and small-world net-

work models, which have aroused the flourishing network science in the past decade. While in the data flood of today, a static connectivity description to such dynamical and evolving complex networks is nevertheless out of the finger to touch the essence. Temporal network gives sufficient respect to categories of temporal information embedded into the connectivity mapping, which covers more than the scopes of time-varying, time-evolving, time-switching networks/systems, etc., since burst dynamics with discontinuous time-stamps, e.g., has never been addressed before. Therefore, as the small tip of a huge iceberg, the mathematic rough-yet definition of temporal network deserves more attention and efforts to formalize its structural description as well as temporal dynamics evolved. In this case, structural controllability of temporal networks may present not only as a new topological metric but also a (potential) bridge to uncover such necessary conditions to achieve optimal performance of controlling temporal networks.

More technical extensions may fall into two branches. On the one hand, the case of multiple controllers with efficient switching strategies comes as a next step of this chapter. Although we have started to touch this question [74], the role of temporal motifs [81] has not been uncovered yet, which has been witnessed its powerful impact to human interactive dynamics and can not stand by to the control problem. On the other hand, leaving aside system dynamics over every node of a temporal network is always invincible to offer the control solution in practice. Combining linear/nonlinear, identical/non-identical system dynamics to a network with temporal topological connectivity, pinning control [7, 33, 40, 65, 67] of such nonlinear temporal networks as well as the pace-maker [32] of heterogenous kuramoto-like temporal networking oscillators leave sufficient difficulty to overcome before facing this challenge.

Acknowledgments This work was partly supported by National Science Foundation for Distinguished Young Scholar of China (No. 61425019), National Natural Science Foundation (No. 61273223), the Research Fund for the Doctoral Program of Higher Education (No. 20120071110029) of China, the Key Project of National Social Science Fund of China (No. 12&ZD18), and Shanghai SMEC-EDF Shuguang Project.

References

1. Barabási, A.L.: The origin of bursts and heavy tails in human dynamics. *Nature* **435**(7039), 207–211 (2005)
2. Barabási, A.L., Albert, R.: Emergence of scaling in random networks. *Science* **286**(5439), 509–512 (1999)
3. Bascompte, J.: Disentangling the web of life. *Science* **325**(5939), 416 (2009)
4. Berners-Lee, T., Hall, W., Hendler, J., et al.: Creating a science of the web. *Science* **313**(5788), 769–771 (2006)
5. Chapman, A., Mesbahi, M.: On strong structural controllability of networked systems: a constrained matching approach. In: Proceedings of the IEEE Conference American Control Conference, pp. 6126–6131 (2013)

6. Chapman, A., Nabi-Abdolyousefi, M., Mesbahi, M.: Controllability and observability of network-of-networks via Cartesian products. *IEEE Trans. Autom. Control* **59**(10), 2668–2679 (2014)
7. Chen, G.: Pinning control and synchronization on complex dynamical networks. *Int. J. Control. Autom.* **12**(2), 221–230 (2014)
8. Commault, C., Dion, J.M.: Input addition and leader selection for the controllability of graph-based systems. *Automatica* **49**(11), 3322–3328 (2013)
9. Cornelius, S.P., Kath, W.L., Motter, A.E.: Realistic control of network dynamics. *Nat. Commun.* **4**, 1942 (2013)
10. Cowan, N.J., Chastain, E.J., Vilhena, D.A., Freudenberg, J.S., Bergstrom, C.T.: Nodal dynamics, not degree distributions, determine the structural controllability of complex networks. *PloS ONE* **7**(6), e38398 (2012)
11. Dodds, P.S., Muhamad, R., Watts, D.J.: An experimental study of search in global social networks. *Science* **301**(5634), 827–829 (2003)
12. Eckmann, J.P., Moses, E., Sergi, D.: Entropy of dialogues creates coherent structures in e-mail traffic. *Proc. Natl. Acad. Sci. USA* **101**(40), 14333–14337 (2004)
13. Erdős, P., Rényi, A.: On random graphs I. *Publ. Math. Debr.* **6**, 290–297 (1959)
14. Estrada, E., Hatano, N., Benzi, M.: The physics of communicability in complex networks. *Phys. Rep.* **514**(3), 89–119 (2012)
15. Granovetter, M.S.: The strength of weak ties. *Am. J. Sociol.* **78**(6), 1360–1380 (1973)
16. Grindrod, P., Parsons, M.C., Higham, D.J., Estrada, E.: Communicability across evolving networks. *Phys. Rev. E* **83**(4), 046120 (2011)
17. Gutiérrez, R., Sendiña-Nadal, I., Zanin, M., Papo, D., Boccaletti, S.: Targeting the dynamics of complex networks. *Sci. Rep.* **2**, 396 (2012)
18. Holme, P.: Network dynamics of ongoing social relationships. *Europhys. Lett.* **64**(3), 427 (2003)
19. Holme, P., Saramäki, J.: Temporal networks. *Phys. Rep.* **519**(3), 97–125 (2012)
20. Hopcroft, J.E., Karp, R.M.: An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. *SIAM J. Comput.* **2**(4), 225–231 (1973)
21. Hufnagel, L., Brockmann, D., Geisel, T.: Forecast and control of epidemics in a globalized world. *Proc. Natl. Acad. Sci. USA* **101**(42), 15124–15129 (2004)
22. Isella, L., Stehlé, J., Barrat, A., Cattuto, C., Pinton, J.F., Van den Broeck, W.: What's in a crowd? Analysis of face-to-face behavioral networks. *J. Theor. Biol.* **271**(1), 166–180 (2011)
23. Jarczyk, J.C., Svaricek, F., Alt, B.: Strong structural controllability of linear systems revisited. In: Proceedings of the Conference CDC-ECE, pp. 1213–1218 (2011)
24. Jia, T., Barabási, A.L.: Control capacity and a random sampling method in exploring controllability of complex networks. *Sci. Rep.* **3**, 2354 (2013)
25. Jia, T., Liu, Y.Y., Csóka, E., et al.: Emergence of bimodality in controlling complex networks. *Nat. Commun.* **4**, 2002 (2013)
26. Kalman, R.E.: Mathematical description of linear dynamical systems. *J. Soc. Ind. Appl. Math. Ser. A* **1**(2), 152–192 (1963)
27. Karsai, M., Kaski, K., Kertész, J.: Correlated dynamics in egocentric communication networks. *Plos ONE* **7**(7), e40612 (2012)
28. Kim, H., Anderson, R.: Temporal node centrality in complex networks. *Phys. Rev. E* **85**(2), 026107 (2012)
29. Kostakos, V.: Temporal graphs. *Phys. A* **388**(6), 1007–1023 (2009)
30. Krings, G., Karsai, M., Bernhardsson, S., et al.: Effects of time window size and placement on the structure of an aggregated communication network. *EPJ Data Sci.* **1**(1), 1–16 (2012)
31. Lee, S., Rocha, L.E., Liljeros, F., Holme, P.: Exploiting temporal network structures of human interaction to effectively immunize populations. *PloS ONE* **7**(5), e36439 (2012)
32. Li, X., Rao, P.C.: Synchronizing a weighted and weakly-connected kuramoto oscillator digraph with a pacemaker. *IEEE Trans. Circuit Syst. I: Regul. Pap.* **62**(3), 899–905 (2015)
33. Li, X., Wang, X., Chen, G.: Pinning a complex dynamical network to its equilibrium. *IEEE Trans. Circuit Syst. I: Regul. Pap.* **51**(10), 2074–2087 (2004)

34. Lin, C.T.: Structural controllability. *IEEE Trans. Autom. Control* **19**(3), 201–208 (1974)
35. Liu, X., Lin, H., Chen, B.M.: Structural controllability of switched linear systems. *Automatica* **49**(12), 3531–3537 (2013)
36. Liu, Y.Y., Slotine, J.J., Barabási, A.L.: Controllability of complex networks. *Nature* **473**(7346), 167–173 (2011)
37. Liu, Y.Y., Slotine, J.J., Barabási, A.L.: Control centrality and hierarchical structure in complex networks. *Plos ONE* **7**(9), e44459 (2012)
38. Liu, Y.Y., Slotine, J.J., Barabási, A.L.: Observability of complex systems. *Proc. Natl. Acad. Sci. USA* **110**(7), 2460–2465 (2013)
39. Lombardi, A., Hörnquist, M.: Controllability analysis of networks. *Phys. Rev. E* **75**(5), 056110 (2007)
40. Lu, W., Li, X., Rong, Z.: Global stabilization of complex networks with digraph topologies via a local pinning algorithm. *Automatica* **46**(1), 116–121 (2010)
41. Luenberger, D.G.: *Introduction to Dynamic Systems: Theory, Models, & Applications*. Wiley, New York (1979)
42. Mayeda, H., Yamada, T.: Strong structural controllability. *SIAM J. Control Optim.* **17**(1), 123–138 (1979)
43. Mesbahi, M., Egerstedt, M.: *Graph Theoretic Methods in Multiagent Networks*. Princeton University Press, Princeton (2010)
44. Menichetti, G., Dall'Asta, L., Bianconi, G.: Network controllability is determined by the density of low in-degree and out-degree nodes. *Phys. Rev. Lett.* **113**(7), 078701 (2014)
45. Milgram, S.: The small world problem. *Psychol. Today* **2**(1), 60–67 (1967)
46. Nepusz, T., Vicsek, T.: Controlling edge dynamics in complex networks. *Nat. Phys.* **8**(7), 568–573 (2012)
47. Nicosia, V., Tang, J., Musolesi, M., et al.: Components in time-varying graphs. *Chaos* **22**(2), 023101 (2012)
48. Pan, Y., Li, X., Zhan, J.: On the priority maximum matching of structural controllability of temporal networks. In: *Proceedings of the 32nd IEEE Conference Chinese Control Conference*, pp. 1164–1169 (2013)
49. Pan, Y., Li, X.: Towards a graphic tool of structural controllability of temporal networks. In: *Proceedings of the IEEE International Symposium on Circuits and Systems*, pp. 1784–1787 (2014)
50. Pan, Y., Li, X.: Structural controllability and controlling centrality of temporal networks. *PLoS ONE* **9**(4), e94998 (2014)
51. Pastor-Satorras, R., Vespignani, A.: *Evolution and Structure of the Internet: A Statistical Physics Approach*. Cambridge University Press, Cambridge (2007)
52. Perra, N., Baronchelli, A., Mocanu, D., et al.: Random walks and search in time-varying networks. *Phys. Rev. Lett.* **109**(23), 238701 (2012)
53. Perra, N., Gonçalves, B., Pastor-Satorras, R., et al.: Activity driven modeling of time varying networks. *Sci. Rep.* **2**, 469 (2012)
54. Reinschke, K.J., Svaricek, F., Wend, H.D.: On strong structural controllability of linear systems. In: *Proceedings of the 31st IEEE Conference on Decision and Control*, pp. 203–208 (1992)
55. Ribeiro, B., Perra, N., Baronchelli, A.: Quantifying the effect of temporal resolution on time-varying networks. *Sci. Rep.* **3**, 3006 (2013)
56. Ruths, J., Ruths, D.: Control profiles of complex networks. *Science* **343**(6177), 1373–1376 (2014)
57. Schweitzer, F., Fagiolo, G., Sornette, D., et al.: Economic networks: The new challenges. *Science* **325**(5939), 422–425 (2009)
58. Sorrentino, F., di Bernardo, M., Garofalo, F., Chen, G.: Controllability of complex networks via pinning. *Phys. Rev. E* **75**(4), 046103 (2007)
59. Su, H., Wang, X., Lin, Z.: Flocking of multi-agents with a virtual leader. *IEEE Trans. Autom. Control* **54**(2), 293–307 (2009)
60. Su, H., Wang, X.: *Pinning Control of Complex Networked Systems*. Springer, Berlin (2013)

61. Sun, J., Motter, A.E.: Controllability transition and nonlocality in network control. *Phys. Rev. Lett.* **110**(20), 208701 (2013)
62. Tang, J., Scellato, S., Musolesi, M., et al.: Small-world behavior in time-varying graphs. *Phys. Rev. E* **81**(5), 055101 (2010)
63. Wang, B., Gao, L., Gao, Y.: Control range: a controllability-based index for node significance in directed networks. *J. Stat. Mech. Theory* **2012**(04), P04011 (2012)
64. Wang, B., Gao, L., Gao, Y., Deng, Y.: Maintain the structural controllability under malicious attacks on directed networks. *Europhys. Lett.* **101**(5), 58003 (2013)
65. Wang, X.F., Chen, G.: Pinning control of scale-free dynamical networks. *Phys. A* **310**(3), 521–531 (2002)
66. Wang, X.F., Li, X., Chen, G.R.: *Network Science: An Introduction*. Higher Education Press, Beijing (2012)
67. Wang, X., Li, X., Lu, J.: Control and flocking of networked systems via pinning. *IEEE Circuits Syst. Mag.* **10**(3), 83–91 (2010)
68. Wang, W.X., Ni, X., Lai, Y.C., Grebogi, C.: Optimizing controllability of complex networks by minimum structural perturbations. *Phys. Rev. E* **85**(2), 026115 (2012)
69. Watts, D.J., Strogatz, S.H.: Collective dynamics of ‘small-world’ networks. *Nature* **393**(6684), 440–442 (1998)
70. Wu, Y., Zhou, C., Xiao, J., Kurths, J., Schellnhuber, H.J.: Evidence for a bimodal distribution in human communication. *Proc. Natl. Acad. Sci. USA* **107**(44), 18803–18808 (2010)
71. Vidal, M., Cusick, M.E., Barabási, A.L.: Interactome networks and human disease. *Cell* **144**(6), 986–998 (2011)
72. Yan, G., Ren, J., Lai, Y.C., Lai, C.H., Li, B.: Controlling complex networks: how much energy is needed. *Phys. Rev. Lett.* **108**(21), 218703 (2012)
73. Yao, P., Li, X.: On structural controllability of complex networks using polar placement. In: Proceedings of the 33rd IEEE Conference on Chinese Control Conference, pp. 2783–2788 (2014)
74. Yao, P., Li, X.: Structural controllability of temporal networks with single switching controller (unpublished)
75. Yuan, Z., Zhao, C., Di, Z., Wang, W.X., Lai, Y.C.: Exact controllability of complex networks. *Nat. Commun.* **4**, 2247 (2013)
76. Zhan, J., Li, X.: Consensus of sampled-data multi-agent networking systems via model predictive control. *Automatica* **49**(8), 2502–2507 (2013)
77. Zhan, J., Li, X.: Flocking of multi-agent systems via model predictive control based on position-only measurements. *IEEE Trans. Ind. Inf.* **9**(1), 377–385 (2013)
78. Zhang, Y.Q., Li, X.: Characterizing large-scale population’s indoor spatio-temporal interactive behaviors. In: Proceedings of the ACM SIGKDD International Workshop on Urban Computing, pp. 25–32 (2012)
79. Zhang, Y.Q., Li, X.: Temporal dynamics and impact of event interactions in cyber-social populations. *Chaos* **23**(1), 01313 (2013)
80. Zhang, Y.Q., Li, X.: When susceptible-infectious-susceptible contagion meets time-varying networks with identical infectivity. *Europhys. Lett.* **108**(2), 28006 (2014)
81. Zhang, Y.Q., Li, X., Xu, J., Vaslavkos, A.V.: Human interactive patterns in temporal networks. *IEEE Trans. Syst. Man Cybern. Syst.* **45**(2), 214–222 (2015)
82. Zhang, Y., Wang, L., Zhang, Y.Q., Li, X.: Towards a temporal network analysis of interactive WiFi users. *Europhys. Lett.* **98**(6), 68002 (2012)

Chapter 14

A General Model for Studying Time Evolution of Transition Networks

Choujun Zhan, Chi K. Tse and Michael Small

Abstract We consider a class of complex networks whose nodes assume one of several possible states at any time and may change their states from time to time. Such networks, referred to as transition networks in this chapter, represent practical networks of rumor spreading, disease spreading, language evolution, and so on. Here, we derive a general analytical model describing the dynamics of a transition network and derive a simulation algorithm for studying the network evolutionary behavior. By using this model, we can analytically compute the probability that (1) the next transition will happen at a given time; (2) a particular transition will occur; (3) a particular transition will occur with a specific link. This model, derived at a microscopic level, can reveal the transition dynamics of every node. A numerical simulation is taken as an “experiment” or “realization” of the model. We use this model to study the disease propagation dynamics in four different prototypical networks, namely, the regular nearest-neighbor (RN) network, the classical Erdős-Renyí (ER) random graph, the Watts-Strogátz small-world (SW) network, and the Barabási-Albert (BA) scalefree network. We find that the disease propagation dynamics in these four networks generally have different properties but they do share some common features. Furthermore, we utilize the transition network model to predict user growth in the Facebook network. Simulation shows that our model agrees with the historical data. The study can provide a useful tool for a more thorough understanding of the dynamics of transition networks.

C. Zhan · C.K. Tse (✉)

Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hong Kong, People’s Republic of China
e-mail: michael.tse@polyu.edu.hk; encktse@polyu.edu.hk

C. Zhan

e-mail: zchoujun2@gmail.com

M. Small

School of Mathematics and Statistics, The University of Western Australia, Perth,
WA 6009, Australia
e-mail: michael.small@uwa.edu.au

14.1 Introduction

Many large-scale real-world networks, such as the Internet, the global aviation network and protein interaction networks, can be described as *complex networks* in which nodes represent individuals or organizations, and links mimic interactions among the nodes [1, 2]. In many cases, each node of the network may assume several states and can change from one state to another. For instance, in a rumor spreading network [3], each node of the network can be regarded as being in one of three possible states, namely, *spreader*, *ignorant* and *stifler*. An *ignorant* who is ignorant of the rumor, can become a *spreader* who actively spreads the rumor, and a *spreader* can finally become a *stifler* who no longer spreads the rumor. Other examples can be found in epidemic spreading networks described by the susceptible-infected-recovered (SIR) model [4–6], the susceptible-infected-susceptible (SIS) model [7, 8], or the susceptible-infected (SI) model [9–11]. Furthermore, complex networks of language evolution [12, 13], the growth of membership-based websites [14], the spread of social behavior [15, 16], cascading failure, and so on, also fall in this category. In this chapter, we refer to this important class of networks as *transition networks*, which exist in many social, biological and communication systems.

For convenience of illustration, we discuss the main idea here using the familiar SIR epidemic spreading model [17, 18]. Moreover, we stress that the proposed model is generally applicable to networks with any connection mechanism, like the SEIR, SIS, language evolution, human behaviour spreading and so on, and in any network setting, such as the Erdős-Renyí random graph (ER), Watts-Strogátz small-world (WS) and Barabási-Albert (BA) scalefree networks. In an SIR model, individuals are categorized as Susceptible, Infected, and Recovered. For instance, under the homogeneous mixing hypothesis assumption [19, 20], a standard deterministic SIR model can be described in terms of the densities of susceptible $S(t)$, infected $I(t)$, and recovered individuals $R(t)$, along with a set of dynamic equations: $dS/dt = \lambda I(t)S(t)$, $dI/dt = -I(t) + \lambda I(t)S(t)$, $dR/dt = I(t)$. The force of infection, λ , is defined as the *per capita rate* at which a susceptible individual becomes infected from an infected individual. The ODEs provide a crude description revealing only the density dynamics $S(t)$, $I(t)$ and $R(t)$, but fail to reveal disease propagation dynamics at any specific node. The homogeneous mixing hypothesis discards possible heterogeneities in the network, and inhomogeneous dynamics cannot be captured with this standard SIR model. Furthermore, the ODEs are deterministic while disease spreading is often subject to non-deterministic processes depending on the nature of transmission. Hence, in principle, a stochastic model is more realistic than a deterministic one. In addition, stochasticity introduces variances and co-variances which influence the behavior, and helps push the system away from the deterministic attractor so that transitions may play a significant role. Also, stochasticity may cause extinction of the epidemic [21].

The use of ODEs is sufficient for homogeneous networks. To analyze disease propagation in general structure networks, Monte-Carlo simulations can be applied. For instance, let $X(t)$, $Y(t)$ and $Z(t)$ represent the number of the susceptible, infected

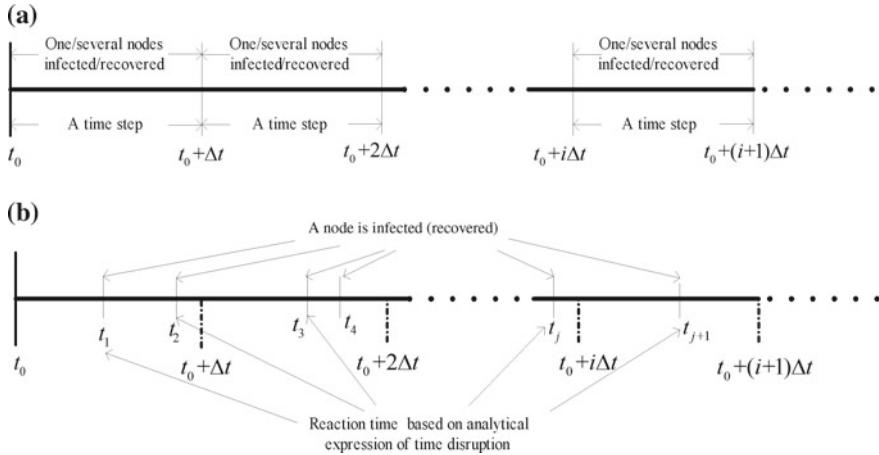


Fig. 14.1 **a** Standard Monte-Carlo simulation where one or more nodes get infected/recovered during one time step; **b** proposed model where only one node is infected/recovered in a time interval

and recovered individuals, respectively. In much of the prior work, a time step with a fixed length is used to simulate disease propagation dynamics in discrete time steps, i.e., at $t = \Delta t, 2\Delta t, \dots$ (or $t = 1, 2, \dots$) [22]. In a time step, one or more individuals are infected or recovered. The simulation procedure generates time series $\{X(t), Y(t), Z(t) \mid t = \Delta t, 2\Delta t, \dots\}$. So, an underlying assumption is that one or more nodes are infected or recovered at the same time $t = \Delta t, 2\Delta t, \dots$ or $t = 1, 2, \dots$, as shown in Fig. 14.1a. Note that the length of time step is always already chosen (fixed) in the simulation. However, in reality, the time when a susceptible node is infected should be distributed over the real time $t \in \mathbb{R}^+$. It is thus more realistic to assume that the time series is $\{X(t), Y(t), Z(t) \mid t \in \mathbb{R}^+\}$, as shown in Fig. 14.1b. In this study, a general model will be provided to tackle this important distinction.

In this chapter, a stochastic process model is established for general complex transition networks with arbitrary connectivity distribution and transition probability. In most cases, the proposed model cannot be solved analytically, as it is the case with most models using ODEs. Hence, the proposed model has to be solved numerically. In our study, a simulation algorithm, which does not involve fixed time steps, is established. Specifically, our model aims to provide analytical expressions of

1. the probability that the next transition will occur at a certain time;
2. the probability that a particular transition will occur;
3. the probability that a particular transition will occur with a specific link.

Clearly, with these expressions, this model can expose the transition dynamics of every individual in continuous time, whereas the ODE-based model describes the dynamics of “density” and the standard Monte-Carlo simulation can only capture the dynamics of each node at pre-defined discrete time-steps. It should be noted that no

mathematical model can fully predict transition dynamics of a natural system. However, the proposed model offers a more realistic physical meaning of the relationship between individuals in a system and is thus able to provide a better overall picture of the transition network behavior.

In addition, our study examines the spreading network in four representative complex network models, namely, the regular nearest-neighbor (RN) network, the ER random-graph network, the WS small-world network, and the BA scale-free network. A brief overview of epidemiological characteristics shows that the epidemic dynamics of the aforementioned four types of complex networks have their own particular characteristics but they also share some common features. One of the findings is that homogeneity or heterogeneity may not be a key condition controlling the rate of epidemic spreading, while the average path length may play an important role. Our findings will be useful for the formulation of epidemic control strategies. For example, identifying and quarantining individuals is an effective measure in ensuring a sufficiently long average path length in a network. Also, the transition network model is used to predict user growth of Facebook during 2004–2013. Simulation shows the proposed model fits well with the historical data. Details of the model derivation and experimental results will be shown in the subsequent sections.

In the next section, we describe the model in detail. In Sect. 14.3, a numerical simulation algorithm is presented. Finally, some numerical simulations of disease propagation and a summary of the results are presented in Sect. 14.4.

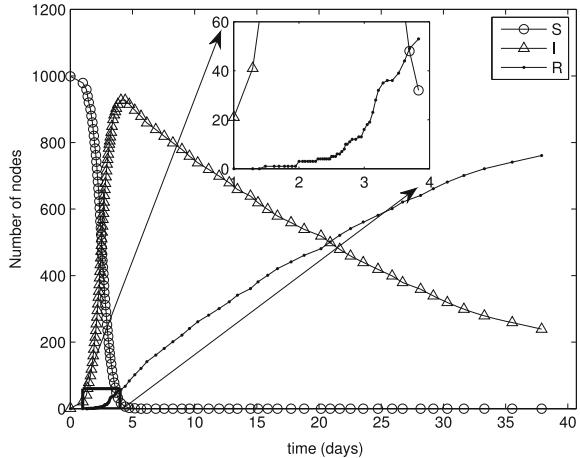
14.2 Formulation of General Stochastic Process Model for Transition Networks

The standard deterministic ODE-based SIR model [23] is established to describe the dynamics of “density”, which is similar to the concept of “molecular concentration” in modeling a chemical reaction process. Hence, while an ODE-based model exposes the dynamics of “molecular concentration” in continuous time, our model reaches to the “molecular” level and reveals the dynamics of every “molecule” in continuous time. To facilitate discussion, a numerical simulation of disease propagation is first provided to give a sense of some differences between the standard SIR model and the proposed model applied to the SIR case. Referring to Fig. 14.1b, a typical numerical simulation using the proposed model generates time series $\{X(t_i), Y(t_i), Z(t_i) \mid t_i \in \mathbb{R}^+, i = 1, 2, \dots\}$, with the following properties:

1. One and only one infection or recovery occurs in the time interval $[t_{i-1}, t_i]$. Thus, we treat the infection/recovery as being completed at time t_i , i.e., at time t_i , the state of the network changes.
2. The numbers of S , I and R individuals are changing at t_i . Hence, in the time interval $[t_{i-1}, t_i]$, we can treat $\{X(t), Y(t), Z(t)\} = \{X(t_i), Y(t_i), Z(t_i)\}$.

As an example, consider the following simulated time series $\{X(t_0) = 999, Y(t_0) = 1, Z(t_0) = 0 \mid t_0 = 0\}$, $\{X(t_1) = 998, Y(t_1) = 2, Z(t_1) = 0 \mid t_0 = 0.541\text{d}\}$,

Fig. 14.2 A profile of disease propagation in a network with entirely susceptible population and a single infectious individual as initial condition



$\{X(t_2) = 998, Y(t_2) = 1, Z(t_2) = 1 \mid t_2 = 2.241d\}, \dots$, where d represents number of days. This data reveals that at $t_0 = 0$, there are 999 susceptible, 1 infected, and 0 recovered individuals, and in the interval $[t_0, t_1] = [0, 0.541d]$, no infection has finished. The first infection completes at $t_1 = 0.541d$, and the numbers of S , I and R change to $X(t_1) = 998, Y(t_1) = 2, Z(t_1) = 0$. Then, there is no infection or recovery until $t_2 = 2.241d$, and an infected individual has recovered during the interval $[t_1, t_2] = [0.541d, 2.241d]$. Hence, with $\{X(t_i), Y(t_i), Z(t_i) \mid t_i \in \mathbb{N}^+, i = 1, 2, \dots\}$, we are able to obtain almost every detail about the spreading of the disease, including when a transition (infection/recovery) occurs, how many transition (infection/recovery) have occurred during a time interval, and when the state of a specific individual is transited (infected/recovery).

Figure 14.2 shows a snapshot of a single simulation of disease propagation in a complex network with 1000 individuals. If a susceptible node is only connected to one infected node, in 1 day, on average, 8 out of 100 susceptible nodes will be infected. At the same time, 6 out of 100 infected nodes will be recovered. The generation of each transition time of the proposed model is based on an analytical expression of the time interval distribution probability. Each simulation will produce different time series. Figure 14.2 shows that most of the susceptible individuals are infected in the first 4 days, which can be regarded as the “outbreak” period of the disease. In the “outbreak” period, the state of the networks changed rapidly, which can be observed in the inset of Fig. 14.2. More detailed analysis will be given in Sect. 14.4.

14.2.1 Preliminaries

Consider a general network $G = (V, E)$, where V and E denote the set of its nodes and edges, respectively. Suppose there are n nodes $\{v_1, v_2, \dots, v_n\}$. Each node can be

in one of k possible states $\{x_1, x_2, \dots, x_k\}$ and can transit from one state to another [24]. This network is defined as *transition network*. The conceptual description of a transition is represented by the *transition channel* as shown below:

$$T_\mu : (x_{p(\mu 1)} - x_{p(\mu 2)} - \cdots - x_{p(\mu L_\mu)}) \xrightarrow{c_\mu} (x_{q(\mu 1)} - x_{q(\mu 2)} - \cdots - x_{q(\mu L_\mu)}), \quad (14.1)$$

where $\mu = 1, 2, \dots, m$ is the index of the transition channel, m is the total number of types of transition channels, T_μ is the μ th transition channel, and L_μ is the number of transition species in channel T_μ . Here, “ $-$ ” represents there are a link. Hence, $x_{p(\mu i)} - x_{p(\mu j)}$ means a node in state $x_{p(\mu i)}$ connects with another node in state $x_{p(\mu j)}$. The arrow indicates the transition direction, with c_μ being the rate of transition. Thus, $(x_{p(\mu 1)} - x_{p(\mu 2)} - \cdots - x_{p(\mu L_\mu)})$ represents the set of prospective transition links before a transition occurs and $(x_{q(\mu 1)} - x_{q(\mu 2)} - \cdots - x_{q(\mu L_\mu)})$ represents the set of resulting links.

Here, we assume that all m transition channels are independent and all prospective transition links of the transition channel T_μ are homogeneous and exclusive. Then, each prospective transition link of the transition channel T_μ has the same transition probability and each transition occurs independently.

The μ th transition channel has a stochastic rate of transition c_μ . Thus, $c_\mu \Delta t$ is the probability that a prospective transition link of channel T_μ at time t will react in the next infinitesimal time interval $(t, t + \Delta t)$. For simplicity, we assume that c_μ is constant, but it can be made time-varying with no significant effect on the analysis.

As an example to illustrate the above definition, we consider a simple epidemiological model of disease spreading in a small network, as shown in Fig. 14.3. Each node of the network represents an individual in its corresponding state (susceptible, infected, or recovered), and each link is a connection along which the infection can spread. In this network, there are 6 nodes, i.e., $V = \{v_1, v_2, \dots, v_6\}$, and 9 edges. Each node has possible 3 states, i.e., $k = 3$ and $\{x_1, x_2, x_3\} = \{S, I, R\}$. At time t_0 , nodes $\{v_1, v_3, v_4, v_5\}$ are susceptible, $\{v_2\}$ is infected and $\{v_6\}$ is recovered. There are $m = 2$ transition channels, namely T_1 and T_2 :

$$T_1 : (S - I) \xrightarrow{c_1} (I - I), \quad (14.2)$$

$$T_2 : (I) \xrightarrow{c_2} (R). \quad (14.3)$$

Here, the first transition channel involves two transition species, and the second channel involves only one species. Thus, $L_1 = 2$ and $L_2 = 1$. For the first transition channel, the set of prospective transition links before transition occurs is $(S - I) = \{e_{2,1}, e_{2,3}, e_{2,4}\}$, and for the second channel, the set of transition links only includes $(I) = \{v_2\}$. Here, we may set $c_1 = 0.04/d$ and $c_2 = 0.01/d$.

Similarly, the dynamics of language competition [12] in social networks can be model as

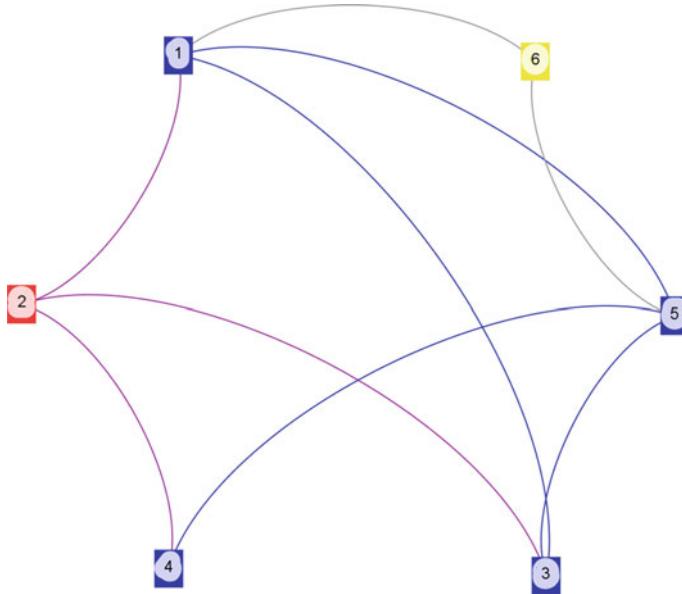


Fig. 14.3 (Color online) A simple network with 6 nodes: red, green and yellow nodes represent infected, suspected and recovered individuals, respectively. Disease can only be spread through the links

$$T_1 : (X - Y) \xrightarrow{c_1} (X - X), \quad (14.4)$$

$$T_2 : (X - Y) \xrightarrow{c_2} (Y - Y). \quad (14.5)$$

where state X represents the state of a speaker who selects language X , while state Y corresponds to the state of a speaker who selects language Y . In this simple model, a speaker can either continue to speak its own language or switch to the other language.

We can also utilize the transition network to model the growth and death of membership-based websites, i.e.,

$$T_1 : (A - I) \xrightarrow{c_1} (A - A), \quad (14.6)$$

$$T_2 : (A) \xrightarrow{c_2} (I), \quad (14.7)$$

$$T_3 : (A - U) \xrightarrow{c_3} (A - A), \quad (14.8)$$

$$T_4 : (U) \xrightarrow{c_4} (A). \quad (14.9)$$

where A represents an active user of a membership website, I is an inactive user, U is a non-member. There are interactions between A , I and U , which forms an online social network.

Now, let us reconsider the simple epidemiological model (14.3). Suppose the network begins to evolve from $t = t_0$, and at time t , the network will be in a

particular state. To determine the state at time t , we have to address the following three questions:

1. When will the next transition occur?
2. What transition will occur?
3. Which one of the prospective transition links will be selected?

Using the above SIR model as an example, the first question essentially means that a time interval Δt should be determined such that one and only one transition will complete at $t = t_0 + \Delta t$. Then, transition channel T_1 or T_2 will be selected for the next transition. Finally, if T_1 is selected (i.e., an individual will be infected), one prospective transition link among $\{e_{2,1}, e_{2,3}, e_{2,4}\}$ will be chosen. To answer the three questions stated above, a general model together with a simulation algorithm is developed in this chapter.

14.2.2 General Stochastic Process Model

The state matrix $\mathcal{S}(t)$ and the number of prospective transition links $h_\mu(\mathcal{S}(t))$ are required for establishing the model.

- $\mathcal{S}(t) \in \Re^{k \times N}$: at time t , if node v_j is in state x_i , $s_{i,j}(t) = 1$; otherwise $s_{i,j}(t) = 0$.
- $h_\mu(\mathcal{S}(t))$: number of distinct prospective transition links with state matrix $\mathcal{S}(t)$ for transition channel T_μ .

Referring to Fig. 14.3, at time t_0 , we have

$$\mathcal{S}(t_0) = \begin{pmatrix} 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}. \quad (14.10)$$

Note that $\mathcal{S}(t)$ includes the state information of all the nodes. Transition channel T_1 includes 3 prospective transition links before transition, i.e., $\{e_{2,1}, e_{2,3}, e_{2,4}\}$. Thus, $h_1(\mathcal{S}(t_0)) = 3$, and likewise, $h_2(\mathcal{S}(t_0)) = 1$.

Let $P[\mathcal{S}(t) = M_s]$ denotes the probability that the transition network is in state $M_s \in \Re^{k \times N}$ at time t . The occurrence of the event $\mathcal{S}(t + \Delta t) = N_s$ can be thought of as an occurrence of the joint event: (1) $(\mathcal{S}(t + \Delta t) = N_s, \mathcal{S}(t) = M_s)$, which means that several transitions are completed in interval Δt ; (2) $(\mathcal{S}(t + \Delta t) = N_s, \mathcal{S}(t) = N_s)$, which means in the time interval Δt , no transition takes place. Therefore, we can write

$$\begin{aligned} P[\mathcal{S}(t + \Delta t) = N_s] &= \sum_{j=1}^{\infty} \sum_{N_s^- \in \Omega_j^-} P[\mathcal{S}(t + \Delta t) = N_s, \mathcal{S}(t) = N_s^-] \\ &\quad + \left(P[\mathcal{S}(t) = N_s] - \sum_{j=1}^{\infty} \sum_{N_s^+ \in \Omega_j^+} P[\mathcal{S}(t + \Delta t) = N_s^+, \mathcal{S}(t) = N_s] \right). \end{aligned} \quad (14.11)$$

where Ω_j^- represents all the possible state matrices N_s^- which can transit into N_s after the j th transition (with $j = 1, 2, \dots, \infty$); also Ω_j^+ represents all the possible state matrices N_s^+ which can transit from N_s after the j th transition (with $j = 1, 2, \dots, \infty$). Here, the probability of $\mathcal{S}(t) = N_s$ at time $t + \Delta t$ is expressed in terms of the probabilities of all the possible states $\mathcal{S}(t)$. Manipulating conditional probabilities, we get

$$\begin{aligned} P[\mathcal{S}(t + \Delta t) = N_s, \mathcal{S}(t) = N_s^-] &= P[\mathcal{S}(t) = N_s^-] \times P[\mathcal{S}(t + \Delta t) = N_s | \mathcal{S}(t) = N_s^-], \\ P[\mathcal{S}(t + \Delta t) = N_s^+, \mathcal{S}(t) = N_s] &= P[\mathcal{S}(t) = N_s] \times P[\mathcal{S}(t + \Delta t) = N_s^+ | \mathcal{S}(t) = N_s]. \end{aligned} \quad (14.12)$$

Putting (14.12) in (14.11), we get

$$\begin{aligned} P[\mathcal{S}(t + \Delta t) = N_s] &= \sum_{j=1}^{\infty} \sum_{N_s^- \in \Omega_j^-} P[\mathcal{S}(t) = N_s^-] \times P[\mathcal{S}(t + \Delta t) = N_s | \mathcal{S}(t) = N_s^-] \\ &\quad + P[\mathcal{S}(t) = N_s] \\ &\quad \times \left(1 - \sum_{j=1}^{\infty} \sum_{N_s^+ \in \Omega_j^+} P[\mathcal{S}(t + \Delta t) = N_s^+ | \mathcal{S}(t) = N_s] \right). \end{aligned} \quad (14.13)$$

Assume that at infinitesimal interval Δt , at most one prospective transition link of one transition channel undergoes a transition. Hence, $\Omega_1^- \neq \emptyset$ and $\Omega_1^+ \neq \emptyset$; otherwise $\Omega_j^- = \emptyset$ and $\Omega_j^+ = \emptyset$ ($j = 2, 3, \dots, \infty$). Thus, (14.13) can be simplified as

$$\begin{aligned} P[\mathcal{S}(t + \Delta t) = N_s] &= \sum_{N_s^- \in \Omega_1^-} P[\mathcal{S}(t) = N_s^-] \times P[\mathcal{S}(t + \Delta t) = N_s | \mathcal{S}(t) = N_s^-] \\ &\quad + P[\mathcal{S}(t) = N_s] \\ &\quad \times \left(1 - \sum_{N_s^+ \in \Omega_1^+} P[\mathcal{S}(t + \Delta t) = N_s^+ | \mathcal{S}(t) = N_s] \right). \end{aligned} \quad (14.14)$$

Since there are m transition channels, (14.14) can be rewritten as

$$\begin{aligned} P[\mathcal{S}(t + \Delta t) = N_s] &= \sum_{\mu=1}^m \sum_{N_s^- \in \Omega_{1,\mu}^-} P[\mathcal{S}(t) = N_s^-] \times P[\mathcal{S}(t + \Delta t) = N_s | \mathcal{S}(t) = N_s^-] \\ &\quad + P[\mathcal{S}(t) = N_s] \\ &\quad \times \left(1 - \sum_{\mu=1}^m \sum_{N_s^+ \in \Omega_{1,\mu}^+} P[\mathcal{S}(t + \Delta t) = N_s^+ | \mathcal{S}(t) = N_s] \right), \end{aligned} \quad (14.15)$$

where $\Omega_1^- = \Omega_{1,1}^- \cup \Omega_{1,2}^- \dots \cup \Omega_{1,m}^-$; $\Omega_1^+ = \Omega_{1,1}^+ \cup \Omega_{1,2}^+ \dots \cup \Omega_{1,m}^+$; $\Omega_{1,\mu}^-$ represents all the state matrices N_s^- which can transit into N_s after a μ th transition; and $\Omega_{1,\mu}^+$ represents all the state matrices N_s^+ which can transit from N_s by a transition of the μ th transition channel.

We assume that the transition probabilities $P[\mathcal{S}(t + \Delta t) = N_s^- | \mathcal{S}(t) = N_s^-]$ and $P[\mathcal{S}(t + \Delta t) = N_s^+ | \mathcal{S}(t) = N_s^-]$ do not explicitly depend upon time t at which the transition occurs. Such transitions are thus *homogenous* with respect to time, but they still depend on Δt . The fact that the transition depends only on the previous step or state is essentially the Markov assumption [25].

Now, recall that $c_\mu \Delta t$ is the probability that a selected prospective transition link of T_μ at time t will make a transition in the next infinitesimal time interval Δt . Then, we have,

$$\begin{aligned} P[\mathcal{S}(t + \Delta t) = N_s^- | \mathcal{S}(t) = N_s^-] &= c_\mu \Delta t, \quad N_s^- \in \Omega_{1,\mu}^- \\ P[\mathcal{S}(t + \Delta t) = N_s^+ | \mathcal{S}(t) = N_s^-] &= c_\mu \Delta t, \quad N_s^+ \in \Omega_{1,\mu}^+. \end{aligned} \quad (14.16)$$

Putting (14.16) in (14.15), we have

$$\begin{aligned} P[\mathcal{S}(t + \Delta t) = N_s] &= \sum_{\mu=1}^m \sum_{N_s^- \in \Omega_{1,\mu}^-} P[\mathcal{S}(t) = N_s^-] \times c_\mu \Delta t \\ &\quad + P[\mathcal{S}(t) = N_s] \times \left(1 - \sum_{\mu=1}^m \sum_{N_s^+ \in \Omega_{1,\mu}^+} c_\mu \Delta t \right). \end{aligned} \quad (14.17)$$

Moreover, since $\sum_{N_s^+ \in \Omega_{1,\mu}^+} 1 = h_\mu(N_s)$, we have

$$\sum_{N_s^+ \in \Omega_{1,\mu}^+} c_\mu \Delta t = c_\mu h_\mu(N_s) \Delta t = P[\mu, \Delta t], \quad (14.18)$$

where $P[\mu, \Delta t]$ is the probability that a T_μ transition will occur in the time interval $(t, t + \Delta t)$.

Now, putting (14.18) in (14.17) yields

$$\begin{aligned} P[\mathcal{S}(t + \Delta t) = N_s] &= \sum_{\mu=1}^m \sum_{N_s^- \in \Omega_{1,\mu}^-} P[\mathcal{S}(t) = N_s^-] c_\mu \Delta t \\ &\quad + P[\mathcal{S}(t) = N_s] \times \left(1 - \sum_{\mu=1}^m c_\mu h_\mu(N_s) \Delta t \right). \end{aligned} \quad (14.19)$$

Finally, re-arranging and taking the limit as $\Delta t \rightarrow 0$, we get the general transition equation as

$$\frac{\partial P[\mathcal{S}(t) = N_s]}{\partial t} = \sum_{\mu=1}^m \sum_{N_s^- \in \Omega_{1,\mu}^-} c_\mu P[\mathcal{S}(t) = N_s^-] - P[\mathcal{S}(t) = N_s] \times \sum_{\mu=1}^m c_\mu h_\mu(N_s). \quad (14.20)$$

14.3 Stochastic Simulation Algorithm

The foregoing derivation has provided a general model for studying the transition dynamics of very general transition networks at the microscopic level. However, as is obvious from (14.20), an exact analytical solution is unlikely to be achieved, and one has to resort to numerical solution. In the 1970s, Gillespie [26, 27] developed a stochastic simulation algorithm to simulate or realize a chemical master equation (CME) model, which is similar to (14.20). In this section, we extend the Gillespie algorithm for simulating transition dynamics of a transition network.

At time t_0 , the network state is $\mathcal{S}(t_0)$. The goal of stochastic simulation is to describe the evolution of state matrix $\mathcal{S}(t)$ from some given initial state $\mathcal{S}(t_0)$. Let $P[\mu, \tau, \Delta\tau]$ represent the probability that transition T_μ will occur in the infinitesimal time interval $(t + \tau, t + \tau + \Delta\tau)$, given the system is in state $\mathcal{S}(t) = N_s$ at time t . The probability $P[\mu, \tau, \Delta\tau]$ is calculated as

$$P[\mu, \tau, \Delta\tau] = P_0(\tau)P[\mu, \Delta\tau], \quad (14.21)$$

where $P_0(\tau)$ is the probability that, given state $\mathcal{S}(t) = N_s$, no transition will occur in the interval $(t, t + \tau)$. From (14.18), we have $P[\mu, \Delta\tau] = c_\mu h_\mu \Delta\tau$. Define $a_\mu = c_\mu h_\mu$, which represents the transition propensity of the T_μ transition. There are m transition channels, from the joint distribution. The probability that any of the m transitions occurs in the interval $\Delta\tau$ is

$$a^* \Delta\tau = \sum_{\mu=1}^m a_\mu \Delta\tau, \quad (14.22)$$

where $a^* = \sum_{\mu=1}^m a_\mu$. Note that the probability that no transitions occur in the interval $\Delta\tau$ is $1 - a^* \Delta\tau$ and therefore

$$P_0(\tau + \Delta\tau) = P_0(\tau) (1 - a^* \Delta\tau). \quad (14.23)$$

Re-arranging and taking the limit $\Delta\tau \rightarrow 0$, we get

$$\frac{dP_0}{d\tau} = -a^* P_0. \quad (14.24)$$

The probability that nothing happens in zero time is one, i.e., $P_0(0) = 1$. Then, the analytical solution of (14.24) is given by

$$P_0(\tau) = P_0(0)e^{-a^*\tau} = e^{-a^*\tau}. \quad (14.25)$$

Combining (14.18), (14.21) and (14.25), we get

$$P[\mu, \tau, \Delta\tau] = P_0(\tau)P[\mu, \Delta\tau] = a_\mu e^{-a^*\tau} \Delta\tau \quad (14.26)$$

which (since $\Delta t \rightarrow 0$) can be written as

$$P[\mu, \tau] = \lim_{\Delta t \rightarrow 0} \frac{P[\mu, \tau, \Delta t]}{\Delta t} = a_\mu e^{-a^* \tau}, \quad (14.27)$$

where $P[\mu, \tau]$ is the probability density that transition T_μ will occur in the infinitesimal time interval $(t_0 + \tau, t_0 + \tau + \Delta\tau)$, namely, the probability that at time t , a transition T_μ occurs in interval $(t_0 + \tau, t_0 + \tau + \Delta\tau)$ and no other transition occurred in the previous interval.

Now, we obtain the probability that the next transition of any type will occur in the interval $(t + \tau, t + \tau + \Delta\tau)$ by integrating (14.27) over all transitions, i.e.,

$$P^{(1)}(\tau) \Delta\tau = \sum_{\mu=1}^m a_\mu e^{-a^* \tau} \Delta\tau = a^* e^{-a^* \tau} \Delta\tau \quad (14.28)$$

$$\Rightarrow P^{(1)}(\tau) = a^* e^{-a^* \tau}, \quad (14.29)$$

where $P^{(1)}(\tau)$ is probability density function (pdf) for the transition time interval, which determines when the next transition will occur. Note that (14.28) gives a *probability for a time interval* $(t_0 + \tau, t_0 + \tau + \Delta\tau)$, but the exact point for the transition to occur is not specified. However, this will give us a means to generate a single-valued time point for occurrence of the next transition. In order to perform simulation, this has to be implemented in an algorithm. At each time step the system is in one state and we can generate a simulation algorithm to answer all the three questions posed earlier based on the analytical expression of (14.27) and (14.28).

14.3.1 Question 1: When Will the Next Transition Occur?

According to $P^{(1)}(\tau) = a^* e^{-a^* \tau}$, the corresponding cumulative distribution function is defined by

$$F(\tau) = \int_{-\infty}^{\tau} P^{(1)}(t) dt = a^* \int_0^{\tau} e^{-a^* t} dt = 1 - e^{-a^* \tau}. \quad (14.30)$$

We can use a random number generator to generate a number r_1 in the unit interval. If we choose a value τ such that $F(\tau) = r_1$, the pdf of τ will be the one corresponding to $P^{(1)}$. The random value of τ can thus be obtained as

$$\tau = F^{-1}(r_1) = \frac{1}{a^*} \ln \left(\frac{1}{1 - r_1} \right), \quad (14.31)$$

which is the time interval corresponding to the next transition.

14.3.2 Question 2: What Transition Will Occur?

The probability that, given a transition occurring at time τ , the transition is of type μ is equal to the conditional probability $P^{(2)}(\mu|\tau)$, which can be readily found as

$$P^{(2)}(\mu|\tau) = \frac{P(\mu, \tau)}{P^{(1)}(\tau)} = \frac{a_\mu e^{-a^* \tau}}{a^* e^{-a^* \tau}} = \frac{a_\mu}{a^*}. \quad (14.32)$$

Then, by generating a second random number r_2 in the unit interval, the type of transition that occurs at time τ will correspond to the value of μ^* that satisfies the inequality

$$\sum_{j=1}^{\mu^*-1} \frac{a_j}{a^*} \leq r_2 < \sum_{j=1}^{\mu^*} \frac{a_j}{a^*}. \quad (14.33)$$

14.3.3 Question 3: Which One of the Prospective Transition Links Will Be Selected?

The question is answered by referring to the probability that the transition links of transition channel T_{μ^*} have the same transition probability. There are $h_{\mu^*}(N_s)$ prospective transition links. Then, the index of transition link is determined by generating a third random number r_3 in the unit interval. The selected transition link corresponds to the value of i^* ($i^* = 1, 2, \dots, h_{\mu^*}(N_s)$), such that the following inequality is satisfied.

$$\frac{i^* - 1}{h_{\mu^*}(N_s)} \leq r_3 < \frac{i^*}{h_{\mu^*}(N_s)}. \quad (14.34)$$

Finally, with the above information, we can update the state matrix $\mathcal{S}(t)$ to $\mathcal{S}(t + \tau)$ and generate the complete dynamical evolution of the transition network.

14.4 Experimental Study of Epidemic Spreading

In this section, several epidemiological characteristics are studied by utilizing the proposed model to generate epidemic dynamics in several complex network settings. Among the many representative network models, the regular nearest-neighbor (RN) network (in which each node is connected to its $2m_0$ neighbors), the classical Erdős-Renyí (ER) random graph [28], the relatively new Watts-Strogatz (WS) small-world networks [29, 30] and the Barabási-Albert (BA) scale-free network [31] are particularly significant and important. In this study, these networks are selected as illustrative examples. For comparison, the four types of networks all have 1000 nodes

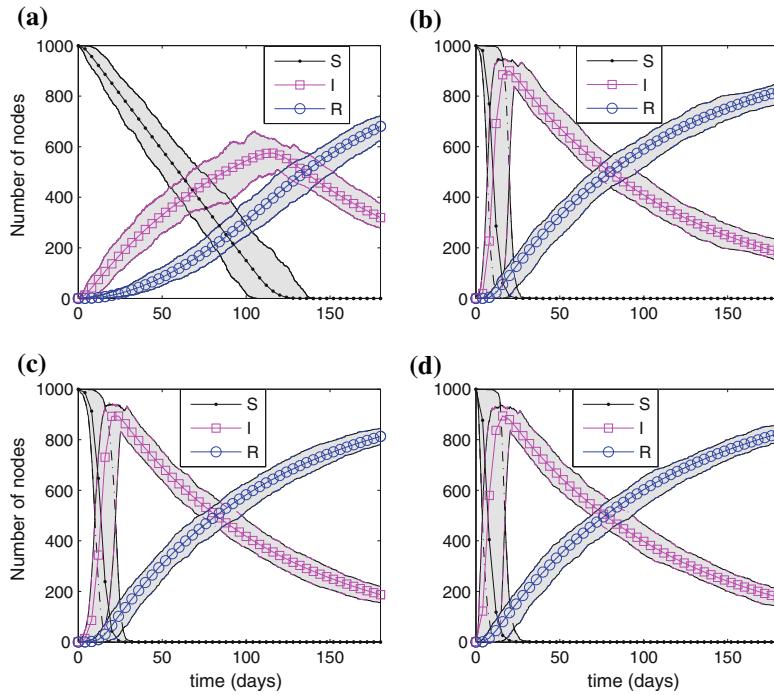


Fig. 14.4 Disease propagation dynamics of the four different networks. *Solid lines* represent the number of *S*, *I* and *R* nodes, and the *gray regions* represent the deviation regions. **a** RN. **b** ER. **c** WS. **d** BA

and about 10000 edges. We set $c_1 = 0.04/d$ and $c_2 = 0.01/d$. All results are obtained by averaging over thousands independent runs for different realizations, based on the four different network models. For each simulation, the model is seeded with one randomly chosen initial infection.

14.4.1 Epidemic Dynamics

Figure 14.4 shows simulation results of the epidemic spreading dynamics for the four networks and show that the range of propagation dynamics in the RN network is wider than the others. The average epidemic spreading dynamics of the four different type networks are plotted in Fig. 14.5a–c, which reveal that the epidemic dynamics of the ER, WS and BA networks have similar behavior, while there is observable difference between the behavior of epidemic dynamics in the RN network and the other three networks.

One important aspect in epidemiology is the “epidemic curve”, which is defined as the number of newly infected cases per time interval, namely, $c(t) = [S(t - \Delta t) -$

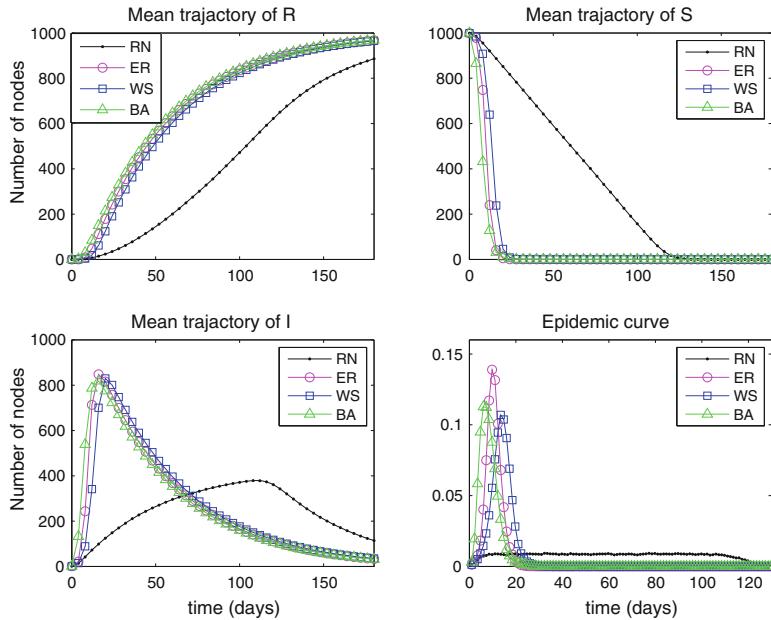


Fig. 14.5 **a** Average of the R dynamics of the four different networks; **b** average of the S dynamics, **c** average of the I dynamics; **d** epidemic curves of the four different networks. y-axis is the normalized infection rate $c(t)$, and x-axis is the time

$S(t)]/R(\infty)$, where Δt is a selected time interval and $R(\infty)$ represents the total number of infected nodes during the spreading. A typical example of the epidemic curve is provided in Fig. 14.5d, which shows the rate of newly infected individuals per day (i.e., $\Delta t = 1$ day). Clearly, the epidemic curves provides information about the time of the peak of the spreading. The figure shows an interesting phenomenon. For epidemics spreading in the ER, WS and BA networks, there are “outbreak” peaks. Specifically, in a very short time, a lot of nodes are infected, while for the RN network, the infection seems more benign.

14.4.2 Spreading Interval Analysis

The simulation algorithm generates a time series $\{X(t_i), Y(t_i), Z(t_i)|i = 1, 2, \dots\}$. At time t_i , a transition (infect or recover) process completes. From this time series, we can obtain $\{X(t_{S,1}), X(t_{S,2}), \dots, X(t_{S,e}), t_{S,i} < t_{S,i+1}, t_i \in \mathbb{R}^+\}$. At $t_{S,i}$, one more node is infected and $X(t_{S,1}) - X(t_{S,2}) = 1$. Similarly, time series $\{Y(t_{I,1}), Y(t_{I,2}), \dots, Y(t_{I,e}) (t_{I,i} < t_{I,i+1}, t_i \in \mathbb{R}^+)\}$ and $\{Z(t_{R,1}), Z(t_{R,2}), \dots, Z(t_{R,e}), t_{R,i} < t_{R,i+1}, t_i \in \mathbb{R}^+\}$ can be found. Here, the definition of spreading period is given as follows. Assume in the time interval $[t_0, t_e]$, a total of N

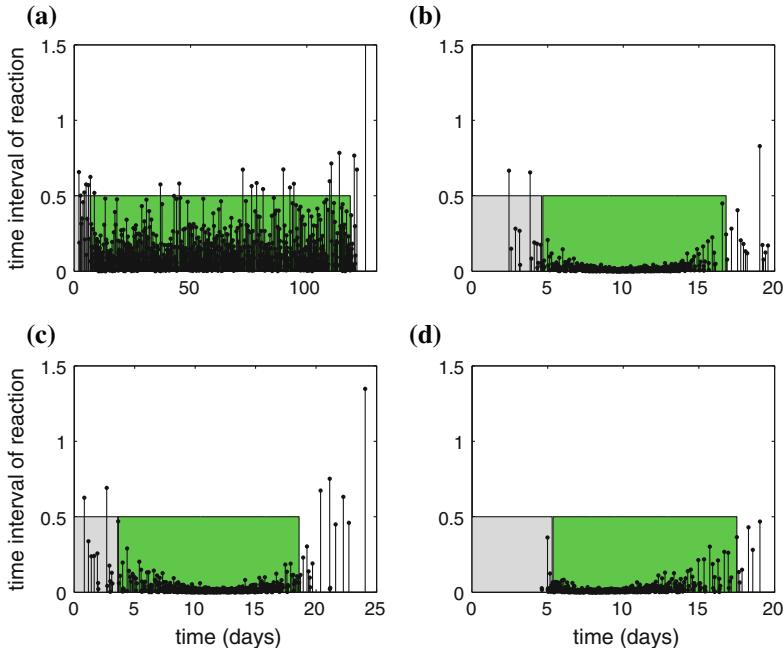


Fig. 14.6 Spreading period of the four type networks. Gray region is outbreak period and green region is incubation period. **a** RN. **b** ER. **c** WS. **d** BA

susceptible nodes are infected, while in time interval $[t_{p1}, t_{pe}]$, there are $N(1 - 2\delta)$ susceptible nodes infected, where $0 < \delta \ll 1$. In intervals $[t_0, t_e]$ and $(t_{pe}, t_e]$, there are only $N\delta$ susceptible nodes infected. In this study, we define $[t_{p1}, t_{pe}]$ as the *outbreak period*. Also, we treat interval $[t_0, t_{p1}]$ as the *incubation period* for the epidemic spreading. There is almost no infection during an incubation period. For instance, for $\delta = 0.005$, 10000 individuals are infected during the first 200 days, but 9900 individuals are infected during day 5 to day 30, while in the interval $[0, 5]$ day and $(30, 200]$ day, only 50 individuals are infected, respectively. Then, the outbreak period is $[5, 30]$ day with a duration of 25 days and the incubation period is $[0, 5]$ day with a duration of 5 days. More precisely, by plotting $\Delta t_{S,i}$ against $t_{S,i}$, we obtain the *spreading interval* profile of the four network models, as given in Fig. 14.6, where the x-axis is $\{t_{S,2}, t_{S,3}, \dots, t_{S,e}, t_{S,i} < t_{S,i+1}, t_i \in \mathbb{R}^+\}$ and the y-axis is $\{\Delta t_{S,i} = t_{S,i+1} - t_{S,i}, i = 1, 2, \dots, e-1, t_i \in \mathbb{R}^+\}$. Each stem reveals that a susceptible node is infected. The thicker the stem, the more susceptible nodes are infected. The gray region indicates the outbreak period whereas the green region indicates the incubation period.

From Fig. 14.6, one can find the outbreak period of the RN network, T_{per}^{RN} , is much longer than the others (T_{per}^{ER} , T_{per}^{BA} and T_{per}^{EG}), which are shown in green. The incubation periods are shown in gray in the same figure. Table 14.1 shows the statistical results of the outbreak periods and the infection periods. It reveals that the WS network

Table 14.1 Average infection period and incubation period (days)

	RN	ER	WS	BA
Incubation period	5.0027 ± 1.6884	4.3943 ± 4.7191	5.0585 ± 1.7402	3.8711 ± 2.5340
Infection period	110.7929 ± 6.2407	12.0122 ± 0.6553	15.1274 ± 1.0332	12.3934 ± 0.7701

has the longest incubation period, while the others have a slightly shorter incubation period. However, the durations of the outbreak periods are very different. In the regular network, it requires about 110 days to infect 99 % of the nodes, while in other networks, the time required is much shorter. Roughly, the outbreak period of the RN network is about 9 times longer than other three networks. Note that the outbreak period of the WS network is about 15 days, which is obviously larger than the outbreak periods of the ER and BA networks which are about 12 days. In conclusion, we have $T_{\text{per}}^{\text{BA}} \approx T_{\text{per}}^{\text{ER}} < T_{\text{per}}^{\text{WS}} \ll T_{\text{per}}^{\text{RN}}$.

From our results, the duration of the outbreak period is related to network structure. Note that RN, ER and WS networks are homogenous, while the BA network is heterogeneous. But the outbreak period relationship is $T_{\text{per}}^{\text{RN}} \gg \{T_{\text{per}}^{\text{BA}}, T_{\text{per}}^{\text{ER}}, T_{\text{per}}^{\text{WS}}\}$. So, homogeneity or heterogeneity is not a key condition that controls the rate of epidemic spreading in a network. Moreover, the approximated average path lengths of the networks [28–30, 32, 33] are found as 1000, 2.4, 30 and 3.5, for the RN, ER, WS and BA networks, respectively. Comparing the relative durations of the outbreak periods, i.e., $T_{\text{per}}^{\text{BA}} \approx T_{\text{per}}^{\text{ER}} < T_{\text{per}}^{\text{WS}} \ll T_{\text{per}}^{\text{RN}}$, it may be inferred that the average path length may be one of the key factors controlling the rate of epidemic spreading in a network. For a network with a shorter average path length, the time required for infecting the whole network is shorter. Further theoretical analysis and simulation results are needed to reveal the relationship between the rate of epidemic spreading and the average path length of the network. In this chapter, we restrict ourselves to the new transition network model and will postpone the detailed study to a future work.

14.5 Modeling Facebook User Growth

Thousands of new products and services emerge every year, and most of them disappear within a few years while a small number of them make their way to the global market with overwhelming success. Facebook, for example, began in December 2004, and there were only about a million users, but after just about 8 years, in January 2013, Facebook users reached 1060 million. The rapid growth of Facebook is an interesting case relevant to the present study of transition networks. Figure 14.7 shows a simple local illustration, where the red nodes represent Facebook users, and the gray nodes are the prospective users who can become Facebook users. Links between two nodes imply that the two individuals have relationship, e.g., being

Fig. 14.7 A simple social network: red nodes represent Facebook users, gray nodes are prospective users who do not join Facebook at the moment

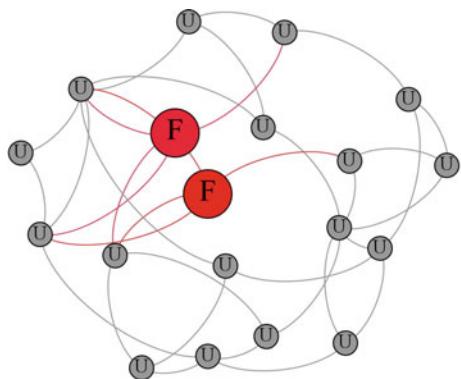
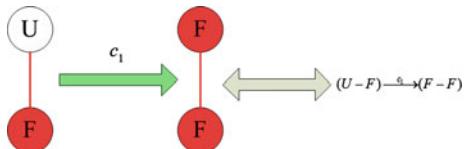


Fig. 14.8 The transition channel of Facebook user growth model



friends, relatives, family members, etc. Each link connecting a Facebook user and a prospective user (red link in Fig. 14.7) is a connection along which the prospective user can “transit” into a Facebook user. This model has a simple practical resemblance as people can be “persuaded” or “infected” by friends and/or family members to become Facebook users. This transition process is shown in Fig. 14.8.

In this network, each node can assume either of two possible states, namely, U and F , which represent a prospective Facebook user and a Facebook user, respectively. Suppose an individual converts from U to F with a probability, per unit of time, of c_1 . Then, there is $m = 1$ transition channel, i.e., T_1 :

$$T_1 : (U - F) \xrightarrow{c_1} (F - F), \quad (14.35)$$

where $(U - F)$ is the set of prospective transition links and $(F - F)$ is the set of resulting links after transition.

To test our model, we collected data of Facebook user growth from 2004–2013 [34], as shown in Fig. 14.9. We fit the model solutions to the data. In this study, for simplicity, one node represents 1 million people, which can be prospective users U or Facebook users F . The networks have 1200 nodes, which form a fully connected network. We set $c_1 = 2.86 \times 10^{-6}/d$. All results are obtained by averaging over thousands independent runs for different realizations. For each simulation, the model is seeded with one randomly chosen node representing one million Facebook users in December 2004. The solid line in Fig. 14.9a shows one realization of the model, which fits well with the historical data of Facebook user growth. Figure 14.9b shows the average of 1000 simulations of Facebook user growth and the deviation. Real-world online social networks may contain millions of nodes. Based on the mean-field

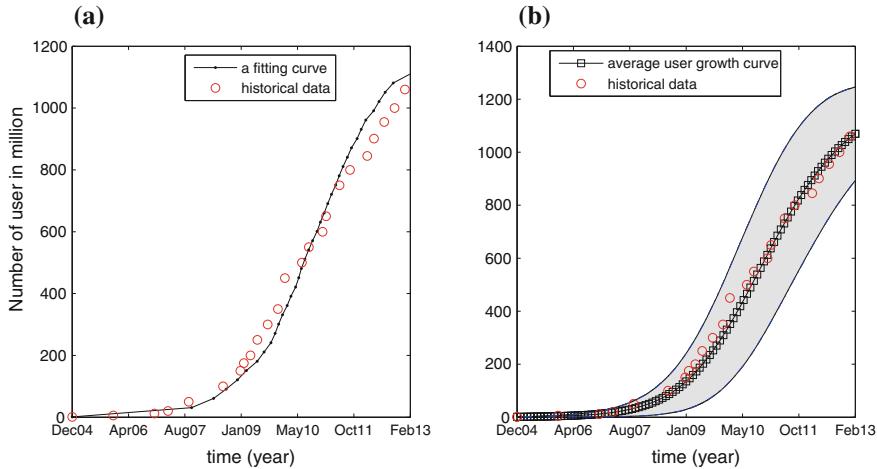


Fig. 14.9 Facebook growth chart during December 2004–2013. **a** Red circles are the historical data of Facebook users, while solid line represents an estimated number of Facebook users; **b** square line represents the average Facebook user growth curve, while the gray sections represent the deviation region

assumption, we can derive an ODE model that describes the user growth profile in continuous time from (14.35). The detailed steps of derivation are omitted for brevity, and we give the final first-order ODE model here:

$$\dot{x}(t) = c_1 N x - c_1 x^2, \quad (14.36)$$

where $x(t)$ is the expected number of Facebook users.

14.6 Conclusion

Based on a stochastic process, we have derived a general model that captures the dynamics of transition networks. A practical simulation algorithm is provided, allowing convenient retrieval of meaningful time profiles for the key quantities that characterize the dynamic behavior of the network. Specifically, we study the epidemiological characteristics of the disease propagation in four representative complex networks, namely, the regular nearest-neighbor network, the random graph, the small-world network, and the scale-free network. By using the proposed model, we avoid making one of the assumptions underlying standard ODE-based models, which is a homogeneous fully connected population, namely, all individuals are susceptible to the disease and all suffer an equal, small, positive probability of contracting the disease. Here, we propose an alternative to the standard SIR model. Unlike the standard SIR model, the proposed model is more realistic and capable of describing dynamics

at a microscopic level, the price to pay being the increased complexity. The results suggest that the topology of the underlying transition networks profoundly affects the performance of epidemic-spreading dynamics. Results may shed light on improving control and even to eradicate the infection from a network, for instance, by carefully selecting and quarantining individuals thereby shortening the average path length which has been found to be related to infection period. Furthermore, we model the growth (spreading) of Facebook with our transition model, and show that our model well fits the historical data of Facebook user growth. Here, the model we developed assumes that the total number of potential users is fixed, and no other competing products come into play. However, in the real world, there are other social networks competing with Facebook. Our transition model can also be modified to accommodate competing transitions, and we will leave this interesting topic to a future study.

In conclusion, the proposed model is expected to find applications in a variety of real-life transition systems, such as rumor spreading, mobile communication, language evolution and grid computing. This model may be applied to communication network for data dissemination, reliable group communication or replicated database maintenance. It will also be helpful in understanding social phenomena such as the spreading of new ideas in population or the efficiency of marketing campaigns. Finally, we point out that the general model studied here can be extended to time-variant transition networks, where the network topology changes with time. We defer a more detailed report of this issue to a future work.

Acknowledgments This work was supported by Hong Kong Polytechnic University Central Research Grant G-YBAT. This work was developed partly during the visit of the second author to the University of Western Australia under the support of the Gledden Visiting Fellowship in 2013.

References

1. Strogatz, S.H.: Exploring complex networks. *Nature* **410**(6825), 268–276 (2001)
2. Albert, R., Barabási, A.L.: Statistical mechanics of complex networks. *Rev. Mod. Phys.* **74**(1), 47 (2002)
3. Daley, D.J., Kendall, D.G.: Epidemics and rumours. *Nature* **204**(4963), 1118 (1964)
4. Newman, M.E.J.: Spread of epidemic disease on networks. *Phys. Rev. E* **66**(1), 016128 (2002)
5. May, R.M., Lloyd, A.L.: Infection dynamics on scale-free networks. *Phys. Rev. E* **64**(6), 066112 (2001)
6. Moreno, Y., Pastor-Satorras, R., Vespignani, A.: Epidemic outbreaks in complex heterogeneous networks. *Eur. Phys. J. B-Condens. Matter Complex Syst.* **26**(4), 521–529 (2002)
7. Pastor-Satorras, R., Vespignani, A.: Epidemic spreading in scale-free networks. *Phys. Rev. Lett.* **86**(14), 3200 (2001)
8. Pastor-Satorras, R., Vespignani, A.: Epidemic dynamics and endemic states in complex networks. *Phys. Rev. E* **63**(6), 066117 (2001)
9. Barthélémy, M., Barrat, A., Pastor-Satorras, R., Vespignani, A.: Velocity and hierarchical spread of epidemic outbreaks in scale-free networks. *Phys. Rev. Lett.* **92**(17), 178701 (2004)
10. Zhou, T., Yan, G., Wang, B.H.: Maximal planar networks with large clustering coefficient and power-law degree distribution. *Phys. Rev. E* **71**(4), 046141 (2005)

11. Vazquez, A.: Polynomial growth in branching processes with diverging reproductive number. *Phys. Rev. Lett.* **96**(3), 038702 (2006)
12. Wang, W.S.Y., Minett, J.W.: The invasion of language: emergence, change and death. *Trends Ecol. Evol.* **20**(5), 263–269 (2005)
13. Ke, J., Gong, T., Wang, W.S.Y.: Language change and social networks. *Comput. Phys. Commun.* **3**(4), 935–949 (2008)
14. Ribeiro, B.: Modeling and predicting the growth and death of membership-based websites. In: Proceedings of 23rd International Conference World Wide Web, International World Wide Web Conferences Steering Committee, pp. 653–664 (2014)
15. Mann, R.P., Faria, J., Sumpter, D.J.T., Krause, J.: The dynamics of audience applause. *J. R. Soc. Interface* **10**(85), 20130466 (2013)
16. Centola, D.: The spread of behavior in an online social network experiment. *Science* **329**(5996), 1194–1197 (2010)
17. Anderson, R.M., May, R.M., Anderson, B.: Infectious Diseases of Humans: Dynamics and Control, vol. 28. Wiley Online Library (1992)
18. Murray, J.D.: Mathematical Biology, vol. 3. Springer, Berlin (1993)
19. Small, M., Tse, C.K.: Small world and scale free model of transmission of SARS. *Int. J. Bifurc. Chaos* **15**(05), 1745–1755 (2005)
20. Small, M., Tse, C.K., Walker, D.M.: Super-spreaders and the rate of transmission of the SARS virus. *Phys. D Nonlinear Phenom.* **215**(2), 146–158 (2006)
21. Keeling, M.J., Rohani, P.: Modeling Infectious Diseases in Humans and Animals. Princeton University Press, Princeton (2008)
22. Zanette, D.H.: Dynamics of rumor propagation on small-world networks. *Phys. Rev. E* **65**(4), 041908 (2002)
23. Moreno, Y., Nekovee, M., Pacheco, A.F.: Dynamics of rumor spreading in complex networks. *Phys. Rev. E* **69**(6), 066130 (2004)
24. Barrat, A., Barthelemy, M., Vespignani, A.: Dynamical Processes in Complex Networks, vol. 1. Cambridge University Press, Cambridge (2008)
25. Øksendal, B.: Stochastic Differential Equations. Springer, Berlin (2003)
26. Gillespie, D.T.: A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *J. Comput. Phys.* **22**(4), 403–434 (1976)
27. Gillespie, D.T.: Exact stochastic simulation of coupled chemical reactions. *J. Phys. Chem.* **81**(25), 2340–2361 (1977)
28. Erdös, P., Rényi, A.: On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci.* **5**, 17–61 (1960)
29. Watts, D.J., Strogatz, S.H.: Collective dynamics of small-world networks. *Nature* **393**(6684), 440–442 (1998)
30. Newman, M.E.J., Watts, D.J.: Renormalization group analysis of the small-world network model. *Phys. Lett. A* **263**(4), 341–346 (1999)
31. Barabási, A.L., Albert, R.: Emergence of scaling in random networks. *Science* **286**(5439), 509–512 (1999)
32. Bollobás, B., Riordan, O.: Mathematical results on scale-free random graphs. *Handb. Graphs Netw.* **1**, 34 (2003)
33. Cohen, R., Havlin, S.: Scale-free networks are ultrasmall. *Phys. Rev. Lett.* **90**(5), 058701 (2003)
34. YahooNews: Number of active users at Facebook over the years (2013)

Chapter 15

Deflection Routing in Complex Networks

Soroush Haeri and Ljiljana Trajkovic

Abstract Deflection routing is a viable contention resolution scheme in bufferless network architectures where contention is the main source of information loss. In recent years, various reinforcement learning-based deflection routing algorithms have been proposed. However, performance of these algorithms has not been evaluated in larger networks that resemble the autonomous system-level view of the Internet. In this Chapter, we compare performance of three reinforcement learning-based deflection routing algorithms by using the National Science Foundation network topology and topologies generated using Waxman and Barabási-Albert algorithms. We examine the scalability of these deflection routing algorithms by increasing the network size while keeping the network load constant.

15.1 Introduction

The Internet is an example of a complex network that has been extensively studied. Optical networks are envisioned to be part of the Internet infrastructure intended to carry high bandwidth backbone traffic. It is expected that optical networks will carry the majority of TCP/IP traffic in the future Internet. Optical burst switching [50] combines the optical circuit switching and the optical packet switching paradigms. In optical burst-switched (OBS) networks, data are optically switched. Optical burst switching offers the reliability of the circuit switching technology and the statistical multiplexing provided by packet switching networks. Statistical multiplexing of bursty traffic enhances the network utilization. Various signaling protocols that have been proposed enable statistical resource sharing of a light-path among multiple traffic flows [6, 19].

Sections of this chapter appeared in conference proceedings and a journal publication: [27–30].

S. Haeri · L. Trajkovic (✉)
School of Engineering Science, Simon Fraser University,
8888 University Drive, Burnaby BC V5A 1S6, Canada
e-mail: ljilja@sfu.ca

In optical burst switching networks, packets are assembled into bursts that are transmitted over optical fibers. Optical burst switching is a buffer-less architecture that has been introduced to eliminate the optical/electrical/optical signal conversions in high-performance optical networks. The optical/electrical conversion occurs when a data packet arrives at a router and the source and destination addresses required for routing need to be extracted from the packet header. The electrical/optical conversion is then required to forward the packet through the appropriate outgoing link. Eliminating the conversions enables high capacity switching with simpler switching architecture and lower power consumption [66]. Furthermore, optical burst switching enables the optical fiber resources to be shared unlike other optical switching technologies such as Synchronous Optical Network (SONET) and Synchronous Digital Hierarchy (SDH) that reserve the entire light-path from a source to a destination [48]. High-speed optical links are often used to connect the Internet Autonomous Systems and, hence, the optical burst switching may be used for inter-autonomous system communications.

Deflection routing is a viable contention resolution scheme that may be employed in buffer-less networks such as networks on chips or OBS networks. It was first introduced as “hot-potato” routing [8] because packets arriving at a node should be immediately forwarded [2, 41]. Contention occurs when according to a routing table, multiple arriving traffic flows at a node need to be routed through a single outgoing link. In this case, only one flow is routed through the optimal link defined by the routing table. In the absence of a contention resolution scheme, the remaining flows are discarded because the optical node possesses no buffers. Instead of buffering or discarding packets of a flow, deflection routing helps to temporarily deflect them away from the path that is prescribed by the routing table. While other methods have also been proposed to resolve contention such as wavelength conversion [39], fiber delay lines [60], and control packet buffering [1], deflection routing has attracted significant attention [15, 37, 38] as a viable method to resolve contention in buffer-less networks. This is because deflection routing requires only software modifications in the routers [67] while other schemes require deployment of special hardware modules.

Deflection routing algorithms proposed in the literature have only been tested on small-size networks such as the National Science Foundation network or general torus topologies that do not resemble the current Internet topologies [11, 29, 30, 34, 57]. Recent insights emanating from the discovery of power-law distribution of nodes degree [23] and scale-free properties of communication networks [4, 7] have influenced the design of routing protocols [40, 59]. Various empirical results confirm the presence of power-laws in the Internet’s inter-autonomous system-level topologies [44, 52, 58]. Waxman [63] and Barabási-Albert [7] algorithms have been widely used to generate Internet-like graphs.

In this chapter, we compare performance of the Q-learning-based Node Degree Dependent (Q-NDD) [30] deflection routing algorithm, the Predictive Q-learning Deflection Routing (PQDR) [29] algorithm, and the Reinforcement Learning-based Deflection Routing Scheme (RLDRS) [11] by using the National Science Foundation (NSF), randomly generated Waxman [63], and scale-free Barabási-Albert [7] network

topologies. Based on these deflection routing algorithms, network nodes learn to deflect packets optimally using reinforcement learning.

The remainder of this Chapter is organized as follows. In Sect. 15.2, we provide a brief survey of deflection routing algorithms. We also describe algorithms for generating network topologies and briefly present the Internet traffic characteristics. The PQDR algorithm and RLDRS are described in Sect. 15.3. The design of Q-NDD deflection routing algorithm follows in Sect. 15.4. Performance evaluation and simulation scenarios are presented in Sect. 15.5. We conclude with Sect. 15.6.

15.2 Related Work

Various routing algorithms that employ reinforcement learning for generating routing policies were proposed in the early days of the Internet development [14, 21, 45, 49]. Routing in communication networks is a process of selecting a path that logically connects two end-points for packet transmission. A common approach is to map the network topology to a weighted graph and set the weight of each edge according to metrics such as number of hops to destination, congestion, latency, link failure, or the business relationships between the edge nodes. The path with the minimum cost is then selected for end-to-end communications.

An agent that learns how to interact with a dynamic environment through trial-and-error may use reinforcement learning techniques for decision-making [33]. Reinforcement learning consists of three abstract phases irrespective of the learning algorithm:

- An agent observes the state of the environment and selects an appropriate action.
- The environment generates a reinforcement signal and transmits it to the agent.
- The agent employs the reinforcement signal to improve its subsequent decisions.

Therefore, a reinforcement learning agent requires information about the state of the environment, reinforcement signals from the environment, and a learning algorithm. Enhancing a node in a network with a reinforcement learning agent that generates deflection decisions requires three components:

- function that maps a collection of the environment variables to an integer (state),
- decision-making algorithm that selects an action based on the state,
- signaling mechanism for sending, receiving, and interpreting the feedback signals.

Q-learning [61] is a simple reinforcement learning algorithm that has been employed for path selection in deflection routing. The algorithm maintains a Q-value $Q(s, a)$ in a Q-table for every state-action pair. Let s_t and a_t denote the encountered state and the action executed by an agent at a time instant t . Furthermore, let r_{t+1} denote the reinforcement signal that the environment has generated for performing

action a_t in state s_t . When the agent receives the reward r_{t+1} , it updates the Q-value that corresponds to the state s_t and action a_t as:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \times \left[r_{t+1} + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t) \right], \quad (15.1)$$

where $0 < \alpha \leq 1$ is the learning rate and $0 \leq \gamma < 1$ is the discount factor. Q-learning has been considered as an approach for generating routing policies. The Q-routing algorithm [14] requires that nodes make their routing decisions locally. Each node learns a local deterministic routing policy using the Q-learning algorithm. Generating the routing policies locally is computationally less intensive. However, the Q-routing algorithm does not generate an optimal routing policy in networks with low loads nor does it learn new optimal policies in cases when network load decreases. Predictive Q-routing [21] addresses these shortcomings by recording the best experiences learned, which may then be reused to predict traffic behavior. The distributed gradient ascent policy search [49], where reinforcement signals are transmitted when a packet is successfully delivered to its destination, has also been proposed for generating optimal routing policies.

Packet routing algorithms in large networks such as the Internet should also consider the business relationships between Internet service providers. Therefore, randomness is not a desired property for a routing algorithm to be deployed in such environment. Consequently, reinforcement learning-based routing algorithms were not widely used for routing in the Internet because of their inherent random structure.

15.2.1 Deflection Routing

Deflection routing is proposed as a viable contention resolution scheme in buffer-less networks such as OBS networks [50]. Two routing protocols operate simultaneously in such networks: an underlying routing protocol such as the Open Shortest Path First (OSPF) that primarily routes packets and a deflection routing algorithm that only deflects packets in case of a contention. Contention occurs when according to the routing table, multiple arriving traffic flows at a node need to be routed through a single outgoing link. In this case, only one flow is routed through the optimal outgoing link defined by the routing table. If no contention resolution scheme is employed, the remaining flows are discarded because the node possesses no buffers. In these cases, deflection routing helps temporarily misroute packets instead of buffering or discarding them.

Slotted and unslotted deflection schemes were compared [13, 20] and performance of a simple random deflection algorithm and loss rates of deflected data were analyzed [12, 26, 64, 71]. Integrations of deflection routing with wavelength conversion and fiber delay lines were also proposed [69, 72]. Deflection routing algorithms generate deflection decisions based on a deflection set, which includes all alternate

links available for deflection. Several algorithms have been proposed to populate large deflection sets while ensuring no routing loops [31, 68].

Performance analysis of deflection routing based on random decisions shows that random deflection may effectively reduce blocking probability and jitter in networks with light traffic loads [56]. Deflection protocols were recently further enhanced by enabling neighboring nodes to exchange traffic information. Hence, each node generates its deflection decisions based on better understanding of its surrounding [24, 47, 57]. Heuristic approaches may also be used to process the information gathered from the neighboring nodes.

Deflection routing may benefit from the random nature of reinforcement learning algorithms. A deflection routing algorithm coexists in the network with an underlying routing protocol that usually generates a significant number of control signals. Therefore, it is desired that deflection routing protocols generate few control signals. Reinforcement learning algorithms enable a deflection routing protocol to generate viable deflection decisions by adding a degree of randomness to the decision making process.

Reinforcement learning techniques have been recently employed to generate deflection decisions. The Q-learning path selection algorithm [34] calculates a priori set of candidate paths $P = \{p_1, \dots, p_m\}$ for tuples (s_i, s_j) , where $s_i, s_j \in S$ and $S = \{s_1, \dots, s_n\}$ denotes the set of all edge nodes in the network. The i th edge maintains a Q-table that contains a quality value (Q-value) for every tuple (s_j, p_k) , where $s_j \in S \setminus \{s_i\}$ and $p_k \in P$. The sets S and P are states and actions, respectively. The Q-value is updated after each decision is made and the score of the path is reduced or increased depending on the received rewards. The algorithm does not specify a signaling method or a procedure for handling feedback signals.

RLDRS [11] employs the Q-learning algorithm for deflection routing. The advantages of RLDRS are its precise signaling and rewarding procedures. Routing algorithms that are based on Q-learning inherit its drawbacks.

The PQDR [29] algorithm employs the predictive Q-routing (PQR) [21] that addresses the shortcomings of Q-learning-based routing by recording the best experiences learned, which may be reused to predict traffic behavior.

A drawback of the Q-learning path selection algorithm, RLDRS, and PQDR is their complexity, which depends on the size of the network. Hence, they are not easily scalable. The recently proposed Q-NDD algorithm [30] employs Q-learning for deflection routing. It scales well in larger networks because its complexity depends on the node degree rather than the network size. In the case of RLDRS and PQDR, nodes receive feedback signals for every packet that they deflect while in the case of Q-NDD, feedback signals are received only if the deflected packet is discarded by another node.

15.2.2 Network Topologies

Many natural and engineering systems have been modeled by random graphs where nodes and edges are generated by random processes. They are referred to as Erdős and Rényi models [22]. Waxman [63] algorithm is commonly used to synthetically generate such random network topologies. In a Waxman graph, an edge that connects nodes u and v exists with a probability:

$$\Pr(\{u, v\}) = \eta \exp\left(\frac{-d(u, v)}{L\delta}\right), \quad (15.2)$$

where $d(u, v)$ is the distance between nodes u and v , L is the maximum distance between the two nodes, and η and δ are parameters in the range $(0, 1]$. Graphs generated with larger η and smaller δ values contain larger number of short edges. These graphs have longer hop diameter, shorter length diameter, and larger number of biconponents [74]. Graphs generated using Waxman algorithm do not resemble the backbone and hierachal structure of the current Internet. Furthermore, the algorithm does not guarantee a connected network [17].

Small-world graphs where nodes and edges are generated so that most of the nodes are connected by a small number of nodes in between were introduced rather recently to model social interactions [62]. A small-world graph may be created from a connected graph that has a high diameter by randomly adding a small number of edges. (The graph diameter is the largest number of vertices that should be traversed in order to travel from one vertex to another.) This construction drastically decreases the graph diameter. Generated networks are also known to have “six degrees of separation.” It has been observed in social network that any two persons are linked by approximately six connections.

Most computer networks may be modeled by scale-free graphs where node degree distribution follows power-laws. Nodes are ranked in descending order based on their degrees. Relationships between node degree and node rank that follow various power-laws have been associated with various network properties. Eigenvalues versus the order index as well as number of nodes within a number of hops versus number of hops also follow various power-laws that have been associated with Internet graph properties [18, 23, 44, 51, 52, 58]. The power-law exponents are calculated from the linear regression lines $10^{(a)}x^{(b)}$, with segment a and slope b when plotted on a log-log scale. The model implies that well-connected network nodes will get even more connected as Internet evolves. This is commonly referred as the “rich get richer” model [7]. Analysis of complex networks also involves discovery of spectral properties of graphs by constructing matrices describing the network connectivity.

Barabási-Albert [7] algorithm generates scale-free graphs that possess power-law distribution of node degrees. It suggests that incremental growth and preferential connectivity are possible causes for the power-law distribution. The algorithm begins with a connected network of n nodes. A new node i that is added to the network connects to an existing node j with probability:

$$\Pr(i, j) = \frac{d_j}{\sum_{k \in N} d_k}, \quad (15.3)$$

where d_j denotes the degree of the node j , N is the set of all network nodes, and $\sum_{k \in N} d_k$ is the sum of all node degrees.

The Internet is often viewed as a network of Autonomous Systems. Groups of networks sharing the same routing policy are identified by Autonomous System Numbers [5]. The Internet topology on Autonomous System-level is the arrangement of autonomous systems and their interconnections. Analyzing the Internet topology and finding properties of associated graphs rely on mining data and capturing information about the Autonomous Systems. It has been established that Internet graphs on the Autonomous System-level exhibit the power-law distribution properties of scale-free graphs [23, 44, 52]. The Barabási-Albert algorithm have been used to generate viable Internet-like graphs.

In 1985, NSF envisioned creating a research network across the United States to connect the recently established supercomputer centers, major universities, and large research laboratories. The NSF network was established in 1986 and operated at 56 kbps. The connections were upgraded to 1.5 Mbps and 45 Mbps in 1988 and 1991, respectively [54]. In 1989, two Federal Internet Exchanges (FIXes) were connected to the NSF network: FIX West at NASA Ames Research Center in Mountain View, California and FIX East at the University of Maryland [42]. The topology of the NSF network after the 1989 transition is shown in Fig. 15.1.

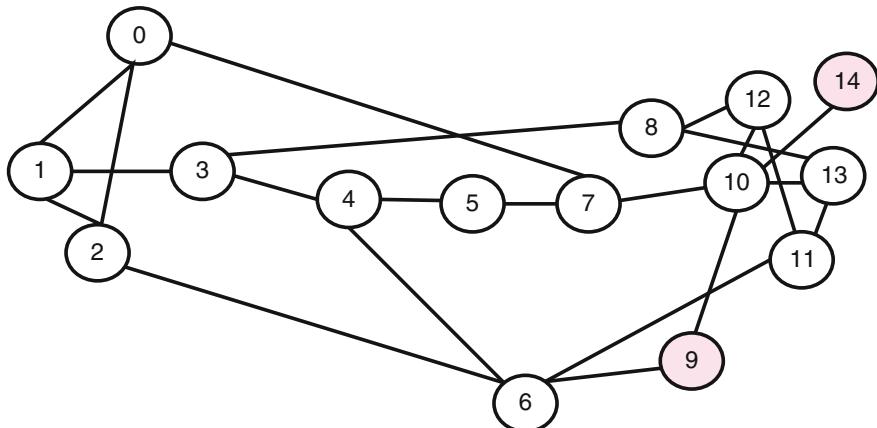


Fig. 15.1 Topology of the NSF network after the 1989 transition. Node 9 and node 14 were added in 1990

15.2.3 Burst Traffic

Simulation of computer networks requires adequate models of network topologies as well as traffic patterns. Traffic measurements help characterize network traffic and are basis for developing traffic models. They are also used to evaluate performance of network protocols and applications. Traffic analysis provides information about the network usage and helps understand the behavior of network users. Furthermore, traffic prediction is important to assess future network capacity requirements used to plan future network developments.

It has been widely accepted that Poisson traffic model that was historically used to model traffic in telephone networks is inadequate to capture qualitative properties of modern packet networks that carry voice, data, image, and video applications [46]. Statistical processes emanating from traffic data collected from various applications indicate that traffic carried by the Internet is self-similar in nature [36]. Self-similarity implies a “fractal-like” behavior and that data on various time scales have similar patterns. Implications of such behavior are: no natural length of bursts, bursts exist across many time scales, traffic does not become “smoother” when aggregated, and traffic becomes more bursty and more self-similar as the traffic volume increases. This behavior is unlike Poisson traffic models where aggregating many traffic flows leads to a white noise effect.

A traffic burst consists of a number of aggregated packets addressed to the same destination. Assembling multiple packets into bursts may result in different statistical characteristics compared to the input packet traffic. Short-range burst traffic characteristics include distribution of burst size and burst inter-arrival time. Two types of burst assembly algorithms may be deployed in OBS networks: time-based and burst length-based. In time-based algorithms, burst inter-arrival times are constant and predefined. In this case, it has been observed that the distribution of burst lengths approaches a Gamma distribution that reaches a Gaussian distribution when the number of packets in a burst is large. With a burst length-based assembly algorithms, the packet size and burst length are predetermined and the burst inter-arrival time is Gaussian distributed. Long-range traffic characteristics deal with correlation structures of traffic over large time scales. It has been reported that long-range dependency of incoming traffic will not change after packets are assembled into bursts irrespective of the traffic load [70].

A Poisson-Pareto burst process has been proposed [3, 75] to model the Internet traffic in optical networks. It may be used to predict performance of optical networks [30]. The burst arrivals are Poisson processes where inter-arrival times between adjacent bursts are exponentially distributed while the burst durations are assumed to be independent and identically distributed Pareto random variables. Pareto distributed burst durations capture the long-range dependent traffic characteristics. Poisson-Pareto burst process has been used to fit the mean, variance, and the Hurst parameter of measured traffic data and thus match the first order and second order statistics.

Traffic modeling affects evaluation of OBS network performance. The effect of the arrival traffic statistics depends on the time scale. Short time scales greatly

influence the behavior of buffer-less high-speed networks. However, self-similarity is negligible when calculating blocking probability even if the offered traffic is long-range dependent over large time scales. Poisson approximation of the burst arrivals provides an upper bound for blocking probability [32]. Hence, we may assume that arrival processes are Poisson. Furthermore, assuming Poisson processes introduces errors that are shown to be within acceptable limits [72].

15.3 PQDR Algorithm and RLDRS

In this section, we present details of the predictive Q-learning deflection routing algorithm (PQDR) [29]. PQDR determines an optimal output link to deflect traffic flows when contention occurs. The algorithm combines the predictive Q-routing (PQR) algorithm [21] and RLDRS [11] to optimally deflect contending flows. When deflecting a traffic flow, the PQDR algorithm stores in a Q-table the accumulated reward for every deflection decision. It also recovers and reselects decisions that are not well rewarded and have not been used over a period of time.

An arbitrary buffer-less network is shown in Fig. 15.2. Let $\mathcal{N} = \{x_1, x_2, \dots, x_n\}$ denote the set of all network nodes. Assume that each node possesses a shortest path routing table and a module that implements the PQDR algorithm to generate deflection decisions. Consider an arbitrary node x_i that is connected to its m neighbors through a set of outgoing links $\mathcal{L} = \{l_0, l_1, \dots, l_m\}$. Node x_i routes the incoming traffic flows f_1 and f_2 to the destination nodes x_{d_1} and x_{d_2} , respectively. According to the shortest path routing table stored in x_i , both flows f_1 and f_2 should be forwarded to node x_j via the outgoing link l_0 . In this case, node x_i forwards flow f_1 through l_0 to the destination x_{d_1} . However, flow f_2 is deflected because node x_i is unable to buffer it. Hence, node x_i employs the PQDR algorithm to select an alternate outgoing link from the set $\mathcal{L} \setminus \{l_0\}$ to deflect flow f_2 . It maintains five tables that are used by PQDR to generate deflection decisions. Four of these tables store statistics for every destination $x \in \mathcal{N} \setminus \{x_i\}$ and outgoing link $l \in \mathcal{L}$:

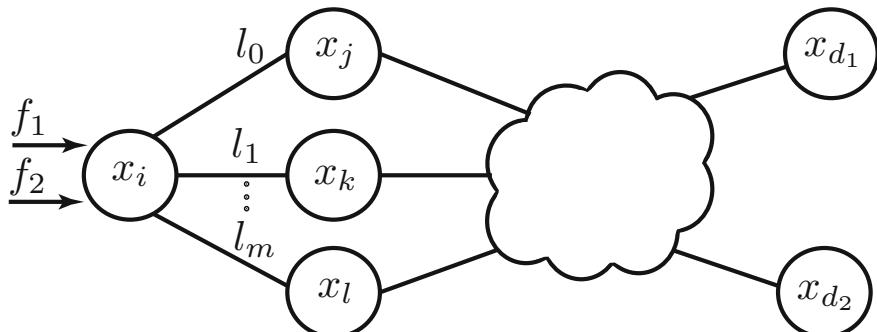


Fig. 15.2 A network with n buffer-less nodes

1. $Q_{x_i}(x, l)$ stores the accumulated rewards that x_i receives for deflecting packets to destinations x via outgoing links l .
2. $B_{x_i}(x, l)$ stores the minimum Q-values that x_i has calculated for deflecting packets to destinations x via outgoing links l .
3. $R_{x_i}(x, l)$ stores recovery rates for decisions to deflect packets to destinations x via outgoing links l .
4. $U_{x_i}(x, l)$ stores the time instant when x_i last updated the (x, l) entry of its Q-table after receiving a reward.

The size of each table is $m \times (n - 1)$, where m and n are the number of elements in the sets \mathcal{L} and \mathcal{N} , respectively. The fifth table $P_{x_i}(l)$ records the blocking probabilities of the outgoing links connected to the node x_i . A time window τ is defined for each node. Within each window, the node counts the successfully transmitted packets λ_{l_i} and the discarded packets ω_{l_i} on every outgoing link $l_i \in \mathcal{L}$. When a window expires, node x_i updates entries in its P_{x_i} table as:

$$P_{x_i}(l_i) = \begin{cases} \frac{\omega_{l_i}}{\lambda_{l_i} + \omega_{l_i}} & \lambda_{l_i} + \omega_{l_i} > 0 \\ 0 & \text{otherwise} \end{cases}. \quad (15.4)$$

The PQDR algorithm needs to know the destination node x_{d_2} of the flow f_2 in order to generate a deflection decision. For every outgoing link $l_i \in \mathcal{L}$, the algorithm first calculates a Δt value as:

$$\Delta t = t_c - U_{x_i}(x_{d_2}, l_i), \quad (15.5)$$

where t_c represents the current time and $U_{x_i}(x_{d_2}, l_i)$ is the last time instant when x_i had received a feedback signal as a result of selecting the outgoing link l_i for deflecting a traffic flow that is destined for node x_{d_2} . The algorithm then calculates $Q'_{x_i}(x_{d_2}, l_i)$ as:

$$Q'_{x_i}(x_{d_2}, l_i) = \max \left(Q_{x_i}(x_{d_2}, l_i) + \Delta t \times R_{x_i}(x_{d_2}, l_i), B_{x_i}(x_{d_2}, l_i) \right). \quad (15.6)$$

$Q_{x_i}(x_{d_2}, l_i)$ is then used to generate the deflection decision (action) ζ :

$$\zeta \leftarrow \arg \min_{l_i \in \mathcal{L}} \{Q'_{x_i}(x_{d_2}, l_i)\}. \quad (15.7)$$

The deflection decision ζ is the index of the outgoing link of node x_i that may be used to deflect the flow f_2 . Let us assume that $\zeta = l_1$ and, therefore, node x_i deflects the traffic flow f_2 via l_1 to its neighbor x_k . When the neighboring node x_k receives the deflected flow f_2 , it either uses its routing table or the PQDR algorithm to forward the flow to its destination through one of its neighbors (x_l). Node x_k then calculates a feedback value v and sends it back to node x_i that had initiated the deflection:

$$v = Q_{x_k}(x_{d_2}, l_{kl}) \times D(x_k, x_l, x_{d_2}), \quad (15.8)$$

where l_{kl} is the link that connects x_k and x_l , $Q_{x_k}(x_{d_2}, l_{kl})$ is the (x_{d_2}, l_{kl}) entry in x_k 's Q-table, and $D(x_k, x_l, x_{d_2})$ is the number of hops from x_k to the destination x_{d_2} through the node x_l . Node x_i receives the feedback v for its action ζ from its neighbor x_k and then calculates the reward r :

$$r = \frac{v \times (1 - P_{x_i}(\zeta))}{D(x_i, x_k, x_{d_2})}, \quad (15.9)$$

where $D(x_i, x_k, x_{d_2})$ is the number of hops from x_i to the destination x_{d_2} through x_k while $P_{x_i}(\zeta)$ is the entry in the x_i 's link blocking probability table P_{x_i} that corresponds to the outgoing link ζ (l_1). The reward r is then used by the x_i 's PQDR module to update the (x_{d_2}, ζ) entries in the Q_{x_i} , B_{x_i} , and R_{x_i} tables. The PQDR algorithm first calculates the difference ϕ between the reward r and $Q_{x_i}(x_{d_2}, \zeta)$:

$$\phi = r - Q_{x_i}(x_{d_2}, \zeta). \quad (15.10)$$

The Q-table is then updated using ϕ as:

$$Q_{x_i}(x_{d_2}, \zeta) = Q_{x_i}(x_{d_2}, \zeta) + \alpha \times \phi, \quad (15.11)$$

where $0 < \alpha \leq 1$ is the learning rate. Table B_{x_i} keeps the minimum Q-values and, hence, its (x_{d_2}, ζ) entry is updated as:

$$B_{x_i}(x_{d_2}, \zeta) = \min(B_{x_i}(x_{d_2}, \zeta), Q_{x_i}(x_{d_2}, \zeta)). \quad (15.12)$$

Table R_{x_i} is updated as:

$$R_{x_i}(x_{d_2}, \zeta) = \begin{cases} R_{x_i}(x_{d_2}, \zeta) + \beta \frac{\phi}{t_c - U_{x_i}(x_{d_2}, \zeta)} & \phi < 0 \\ \gamma R_{x_i}(x_{d_2}, \zeta) & \text{otherwise} \end{cases}, \quad (15.13)$$

where t_c denotes the current time and $0 < \beta \leq 1$ and $0 < \gamma \leq 1$ are recovery learning and decay rates, respectively. Finally, the PQDR algorithm updates table U_{x_i} with current time t_c as:

$$U_{x_i}(x_{d_2}, \zeta) = t_c. \quad (15.14)$$

Signaling algorithms implemented in RLDRS and PQDR are similar. Their main difference is in the learning algorithm. RLDRS uses the Q-learning algorithm and, therefore, it only stores a Q-table $Q_{x_i}(x, l)$ that records the accumulated rewards that the node x_i receives for deflecting packets to destinations x via outgoing links l . As a result, a deflection decision ζ is generated using only the Q-table. Hence, instead of 15.6 and 15.7, RLDRS generates a deflection decision using:

$$\zeta \leftarrow \arg \max_{l_i \in \mathcal{L}} \{Q_{x_i}(x_{d_2}, l_i)\}. \quad (15.15)$$

15.4 Q-NDD Deflection Routing Algorithm

We describe here the newly proposed NDD signaling algorithm [30] and the messages that need to be sent across the network in order to enhance an OBS node with decision-making ability. The NDD algorithm provides a signaling infrastructure that an OBS node may require in order to learn and optimally deflect the bursts in an OBS network.

The flowchart of the signaling algorithm is shown in Fig. 15.3. We consider an OBS network with n nodes. Each network node maintains a Q-table and all nodes are NDD compatible. A burst header that contains the control information associated with a burst is transmitted ahead of the burst. The burst header messages received by a node are passed to the NDD module. The module inspects the routing table for the next hop and then checks the status of the optical interfaces. If the desired optical interface is available, the optical cross-connects are configured according to the path defined by the routing table. If the interface is busy and the burst has not been deflected earlier by any other node, the current states of the optical interfaces and the output port defined by the routing table are passed to the Q-learning module. The states of the optical interfaces are mapped to an ordered string of 0s and 1s, where idle and busy interfaces are denoted by 0 and 1, respectively. We refer to the information passed to the Q-learning module as a *state*. The Q-learning module inspects the Q-table entry for the current *state*. If there is an entry, the learning module selects for deflection the output port that is associated with the maximum Q-value. However, if the learning module is unable to find a Q-table entry for the encountered state, it first initializes an entry for that *state* by assigning uniformly drawn random Q-values to all possible actions and then selects the action with the maximum Q-value. The Q-learning module returns to the NDD module the best selected output port for burst deflection. The following information is then added to the burst header:

- a unique *ID* number used to identify the feedback message that pertains to a deflection,
- the address of the node that initiated the deflection, to be used by other nodes as the destination for the feedback messages,
- a deflection hop counter *DHC*, which is incremented each time other nodes deflect the burst.

When a burst is to be deflected at a node for the first time, the node records the current time as the deflection time *DfT* along with the *ID* assigned to the burst. The Drop Notification (DN) timer is initiated and the burst is deflected to the port that is selected by the Q-learning module. A maximum value for the DN timer is set to DN_{max} , which indicates expiration of the timer. The purpose of this timer is to reduce the number of feedback signals.

After a decision is made to perform a deflection, the Q-learning module waits for the feedback. It makes no new decisions during the idle interval. The deflected burst is discarded when either:

- its *DHC* reaches the maximum permissible number of deflections DHC_{max} ,
- it reaches a fully congested node.

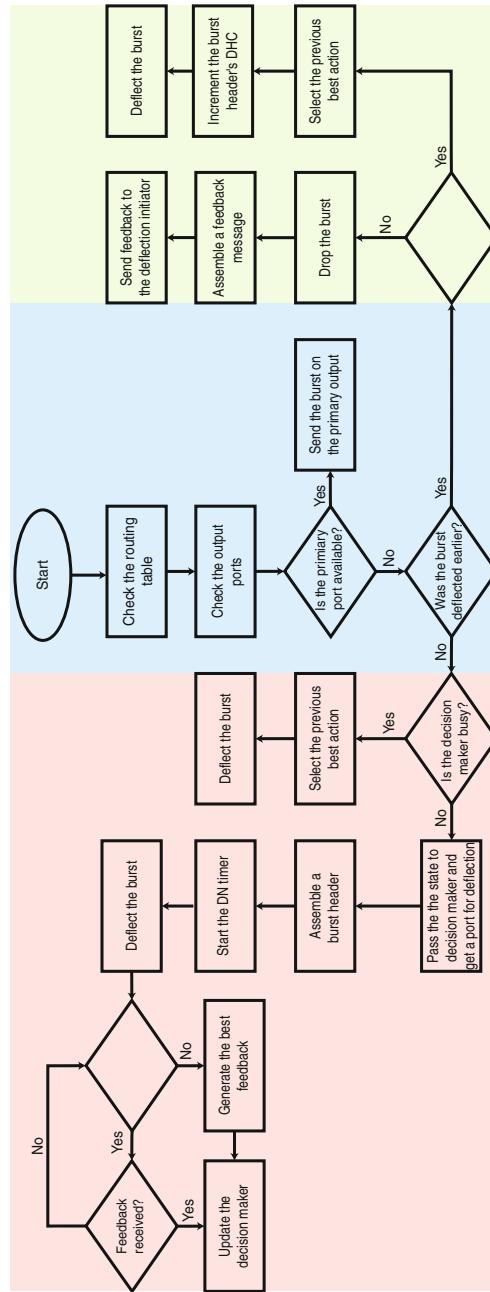


Fig. 15.3 The flowchart of the proposed signaling algorithm. The DN timer denotes the drop notification timer. Nodes wait for feedback signals until this timer reaches DHC_{max} . DHC denotes the deflection hop counter. This counter is a field in the burst header that is incremented by one each time the burst is deflected. DHC_{max} is set in order to control the volume of deflected traffic. A burst is discarded when its DHC value reaches the maximum

The node that discards the deflected burst assembles a feedback message composed of the burst ID , DHC , and the time instant when the burst was discarded (drop time DrT). The feedback message is then sent to the node that initiated the deflection.

When the node that initiated the deflection receives the feedback message, it calculates the total travel time TTT that the burst has spent in the network after the first deflection:

$$TTT = DrT - DfT. \quad (15.16)$$

The TTT and DHC values are then used by the Q-learning module to update its statistics. If no feedback message is received until the DN timer expires, the node assumes that the burst has arrived successfully to its destination. The node may then update its learning module with the reinforcement signal that contains $TTT = 0$ and $DHC = 0$. A decreasing function with the global maximum at $(0, 0)$ may be used as a reward function to map TTT and DHC to a real value r . The Q-learning module updates the Q-value of the current state and the selected action as:

$$Q(s, a) \leftarrow Q(s, a) + \alpha(r - Q(s, a)). \quad (15.17)$$

An OBS node records the best action selected by the Q-learning module. These records are used if a node needs to deflect a burst:

- that has been deflected earlier
- or
- during an idle interval.

In order to reduce the excess traffic generated by the number of feedback messages, a node receives feedback messages only when it deflects bursts that have not been deflected earlier. Hence, deflecting a burst that has been deflected earlier does not enhance the node's decision-making ability.

15.5 Simulation Results

In order to evaluate and compare performance of Q-NDD, PQDR, and RLDRS, we implement these algorithms in the ns-3 network simulator [55] using the iDef framework [27]. ns-3 [55] is a discrete-event network simulator that is publicly distributed under the GNU GPLv2 [25] license. iDef [27] is designed to facilitate development of reinforcement learning-based deflection routing protocols by using its mapping, decision-making, and signaling modules. iDef is designed to minimize the dependency among its modules. Its components are shown in Fig. 15.4.

We compare the algorithms based on burst loss probability, number of deflections, average end-to-end delay, and average number of hops traveled by bursts. We first use the National Science Foundation (NSF) network topology shown in Fig. 15.1,

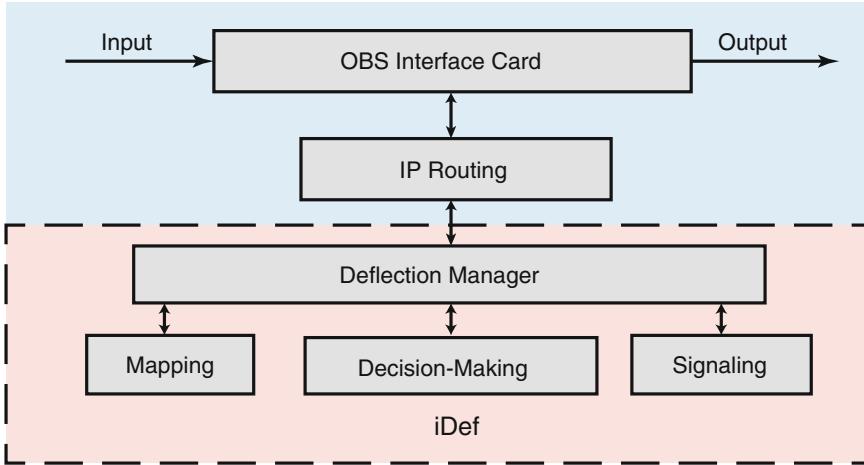


Fig. 15.4 iDef building blocks: The iDef is composed of deflection manager, mapping, signaling, and decision-making modules. The deflection manager module coordinates the communication between modules. Its purpose is to remove dependency among modules

which has been extensively used to evaluate performance of OBS networks [9–11, 34, 38, 64, 65]. We also use the Boston University Representative Internet Topology Generator (BRITE) [16] to generate autonomous system-level topologies that consist of 10, 20, 50, 100, 200, 500, and 1,000 nodes. These topologies are generated using the Waxman and Barabási-Albert algorithms. In all simulation scenarios, we allow up to two deflections per burst ($DHC_{max} = 2$). The burst header processing time is set to 0.1 ms.

15.5.1 NSF Network Simulation Scenario

The NSF network topology was generated by extracting the geodetic coordinates [35] of the NSF network nodes from the Google Earth [53]. They are then transformed to the Cartesian coordinates. We assume the buffer-less OBS architecture for data transmission where nodes are connected using bi-directional 1 Gbps fiber links with 8 or 64 wavelengths.

Multiple Poisson traffic flows with a data rate of 1 Gbps are transmitted randomly across the network. Each Poisson flow is 50 bursts long with each burst containing 12.5 kB of payload. While the burst arrival process depends on the aggregation algorithm [43] deployed in a node, the Poisson process has been widely used for performance analysis of OBS networks because it is mathematically tractable [70, 73]. Each simulation scenario is repeated five times with various random assignments of nodes as sources and destinations. Simulation results are averaged over five runs.

The duration of the sliding window that the nodes employ to calculate burst loss probability on each of their interfaces is set to $\tau = 50$ ms. The learning, learning recovery, and recovery decay rates are $\alpha = 0.1$, $\beta = 0.7$, and $\gamma = 0.9$, respectively [21].

The burst loss probability as a function of the number of Poisson flows for 8 and 64 wavelengths is shown in Fig. 15.5. In all scenarios, the PQDR algorithm performs better than Q-NDD and RLDRS in terms of burst loss probability. The results show that PQDR scales better as the number of wavelengths increases. For example, PQDR performs on average 7.0 % better than RLDRS in case of 8 wavelengths and 800 traffic flows, as shown in Fig. 15.5 (*top*). Simulation results shown in Fig. 15.5 (*bottom*) illustrate that PQDR performs on average 10 % better than RLDRS in case of 64 wavelengths and 5,000 traffic flows.

Although burst deflection reduces the burst loss probability, it introduces excess traffic load to the network. This behavior is undesired from the traffic engineering point of view. Therefore, the volume of the deflected traffic should also be considered as a performance measure for deflection routing algorithms. We use two metrics to compare the Q-NDD algorithm, PQDR algorithm, and RLDRS in terms of the volume of the deflected traffic: deflection ratio and average number of deflections. We define deflection ratio as the number of deflected bursts divided by the number of transmitted bursts:

$$\text{Deflection ratio} = \frac{\text{Number of deflected bursts}}{\text{Number of transmitted bursts}}. \quad (15.18)$$

Deflection ratio and average number of deflections as a function of the number of traffic flows are shown in Fig. 15.6 and Fig. 15.7, respectively. PQDR deflects on average 20 % fewer bursts than Q-NDD and RLDRS in the 64-wavelength scenario, as shown in Fig. 15.7.

Performance of the PQDR algorithm, the Q-NDD algorithm, and RLDRS in terms of average number of hops for the 64-wavelength scenario is shown in Fig. 15.8. When generating the reward signals that are used to update Q-values, the PQDR algorithm and RLDRS consider the number of hops to the destination. Hence, they perform better than Q-NDD in terms of the average number of hops traveled by bursts. In addition to Q-values, PQDR utilizes other variables to generate deflection decisions, which may result in selection of longer paths. Therefore, bursts may travel on average through additional hops.

Performance of the PQDR algorithm, the Q-NDD algorithm, and RLDRS in terms of average end-to-end delay for the 64-wavelength scenario is shown in Fig. 15.9. Bursts experience smaller end-to-end delays in case of RLDRS because the scheme maintains only one table (Q-table) and, therefore, the link selection process and table updates are faster than in the case of PQDR.

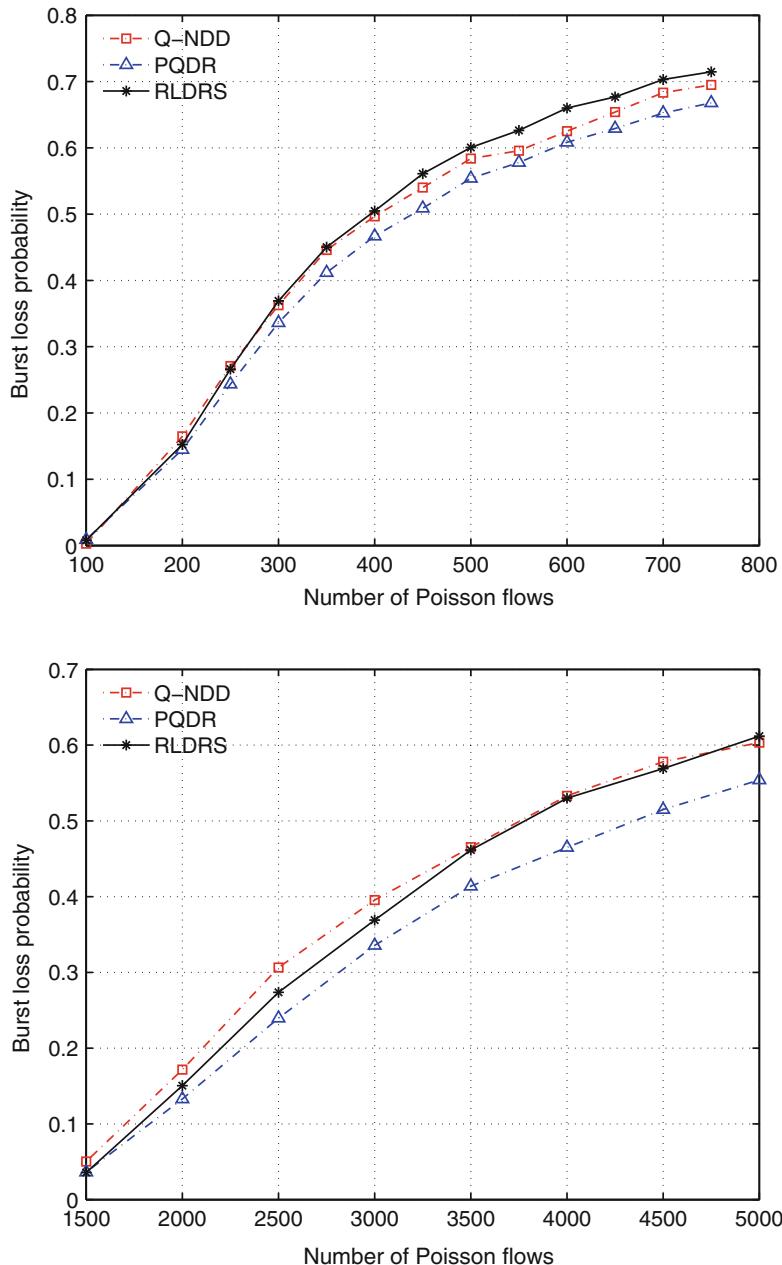


Fig. 15.5 Burst loss probability as a function of the number of Poisson flows in the NSF network simulation scenario with 8 wavelengths (*top*) and 64 wavelengths (*bottom*)

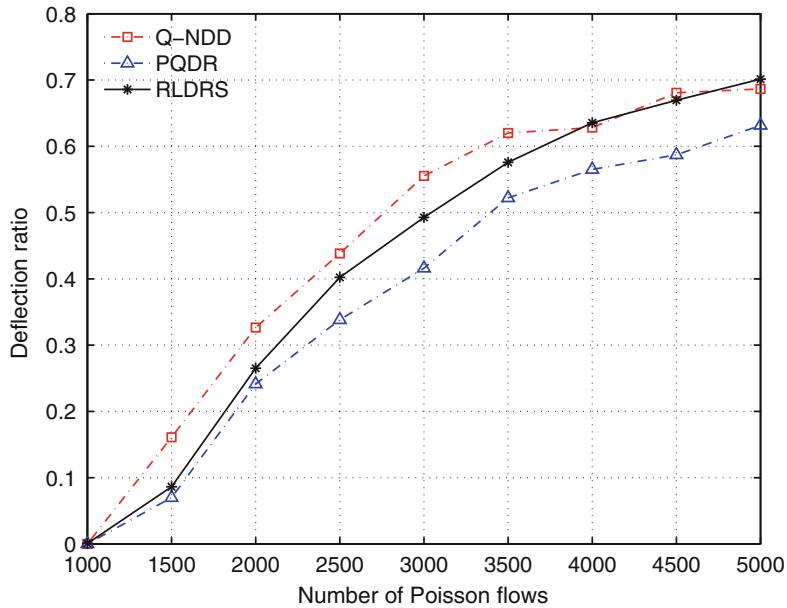


Fig. 15.6 Deflections ratio as a function of the number of traffic flows in the NSF network scenario with 64 wavelengths. PQDR deflects fewer packets

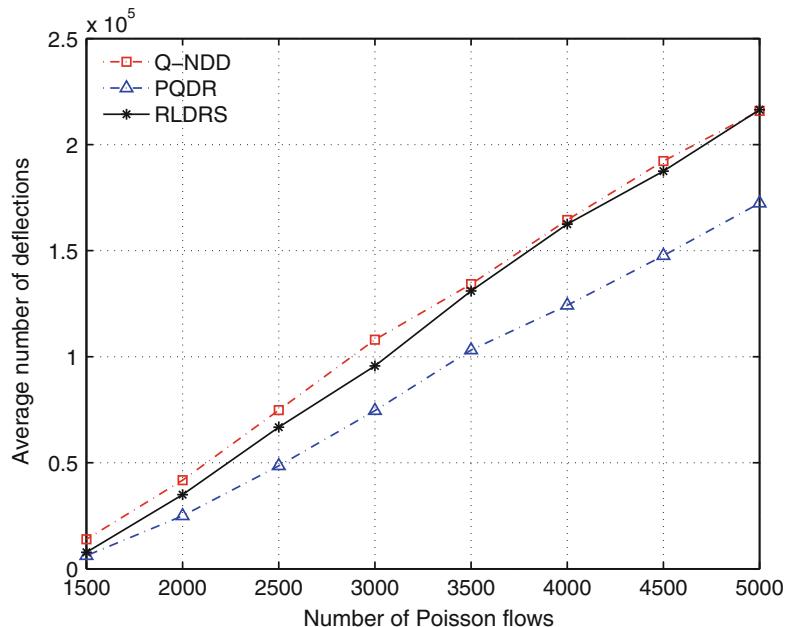


Fig. 15.7 Average number of deflections as a function of the number of traffic flows in the NSF network scenario with 64 wavelengths. PQDR deflects fewer packets

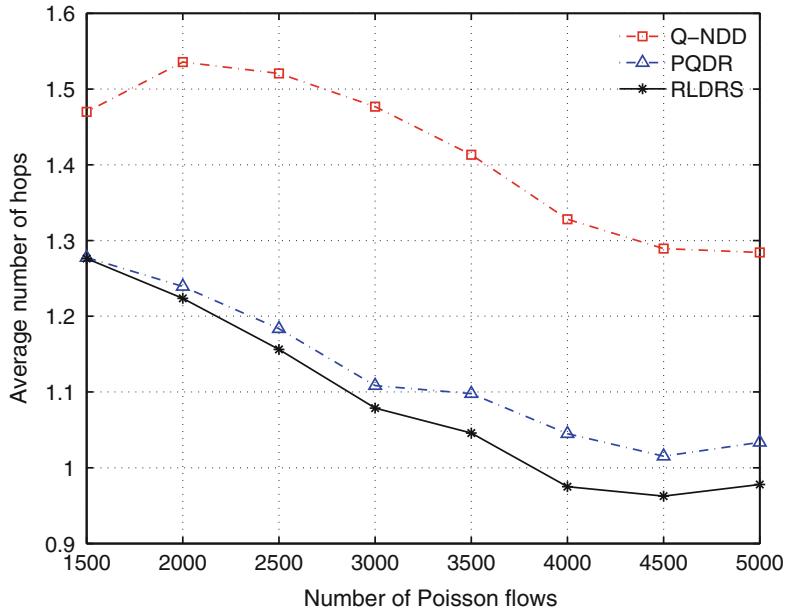


Fig. 15.8 Average number of hops traveled by bursts as a function of number of flows in the NSF network scenario with 64 wavelengths. In the case of RLDRS, the bursts travel the least number of hops

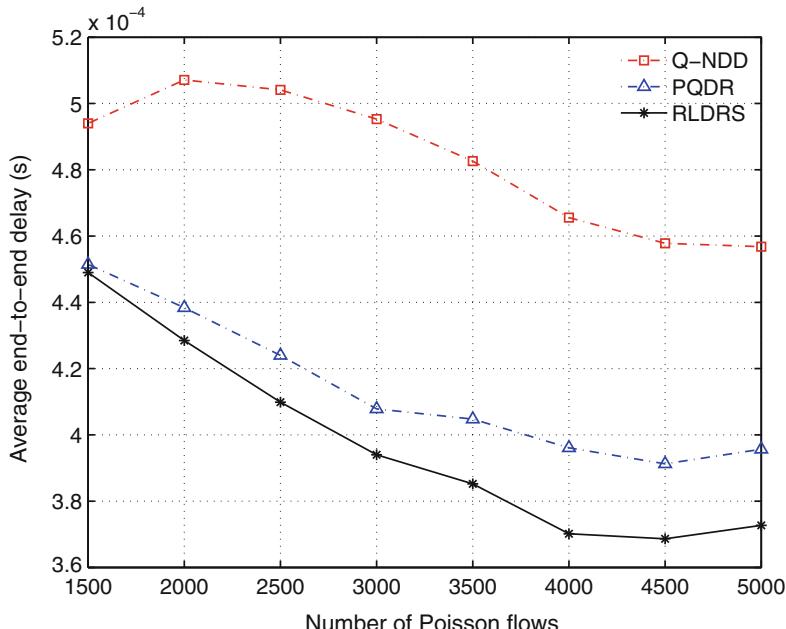


Fig. 15.9 Average end-to-end delay as a function of number of flows in the NSF network scenario with 64 wavelengths. In the case of RLDRS, the bursts experience the smallest end-to-end delay

Table 15.1 Summary of simulation scenarios

Topology generating algorithm	Deflection routing algorithm	Number of nodes	Number of links	Number of flows
Waxman Barabási-Albert	Q-NDD PQDR RLDRS	10	30	24
		20	60	48
		50	150	120
		100	300	240
		200	600	480
		500	1,500	1,200
		1,000	3,000	2,400

15.5.2 Waxman and Barabási-Albert Topologies

We consider a buffer-less OBS architecture for data transmission where nodes are connected using bi-directional 1 Gbps single wavelength fiber links. The nodes are randomly placed and each node is connected to three other nodes. Subsets of nodes are randomly selected as sources and destinations of Poisson traffic flows. Multiple Poisson flows with a data rate of 0.5 Gbps are transmitted across the network. Each Poisson flow is 50 bursts long with each burst containing 12.5 kB of payload. Each simulation scenario is repeated two times with various random assignments of nodes as sources and destinations. Simulation results are averaged over two simulation runs. For each scenario, the network load is maintained at 20%. Hence, scenarios with 10, 20, 50, 100, 200, 500, and 1,000 nodes have 24, 48, 120, 240, 480, 1,200, and 2,400 Poisson traffic flows, respectively. Simulations were performed on a Dell Optiplex-790 with 16 GB memory and the Intel Core i7 2600 processor. Simulation scenarios are shown in Table 15.1.

15.5.3 Burst-Loss Probability

Performance of Q-NDD, PQDR, and RLDRS in terms of burst-loss probability as a function of number of nodes for Waxman and Barabási-Albert network topologies is shown in Fig. 15.10. Burst-loss probability has a logarithmic trend. It is slightly higher in Barabási-Albert networks. Q-NDD scales better than PQDR and RLDRS as the size of the network grows.

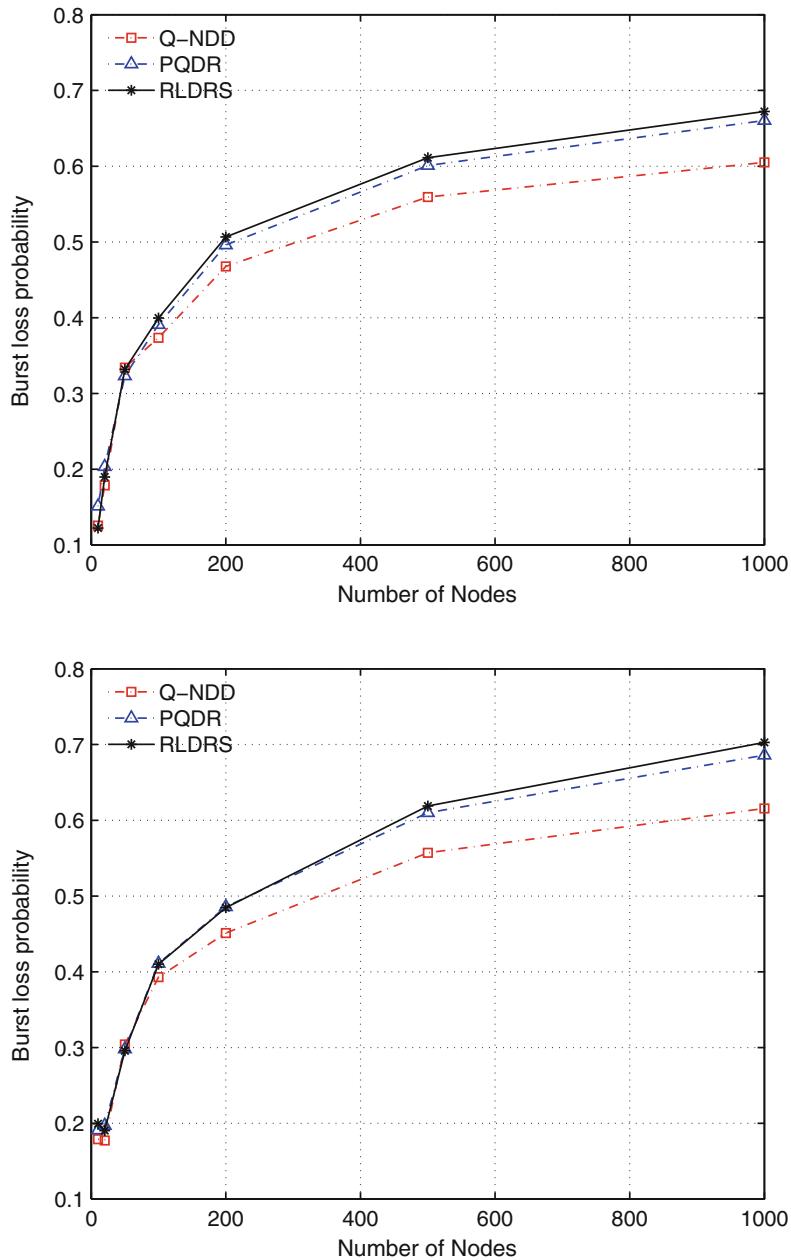


Fig. 15.10 Burst loss probability as a function of the number of nodes in Waxman (top) and Barabási-Albert (bottom) networks

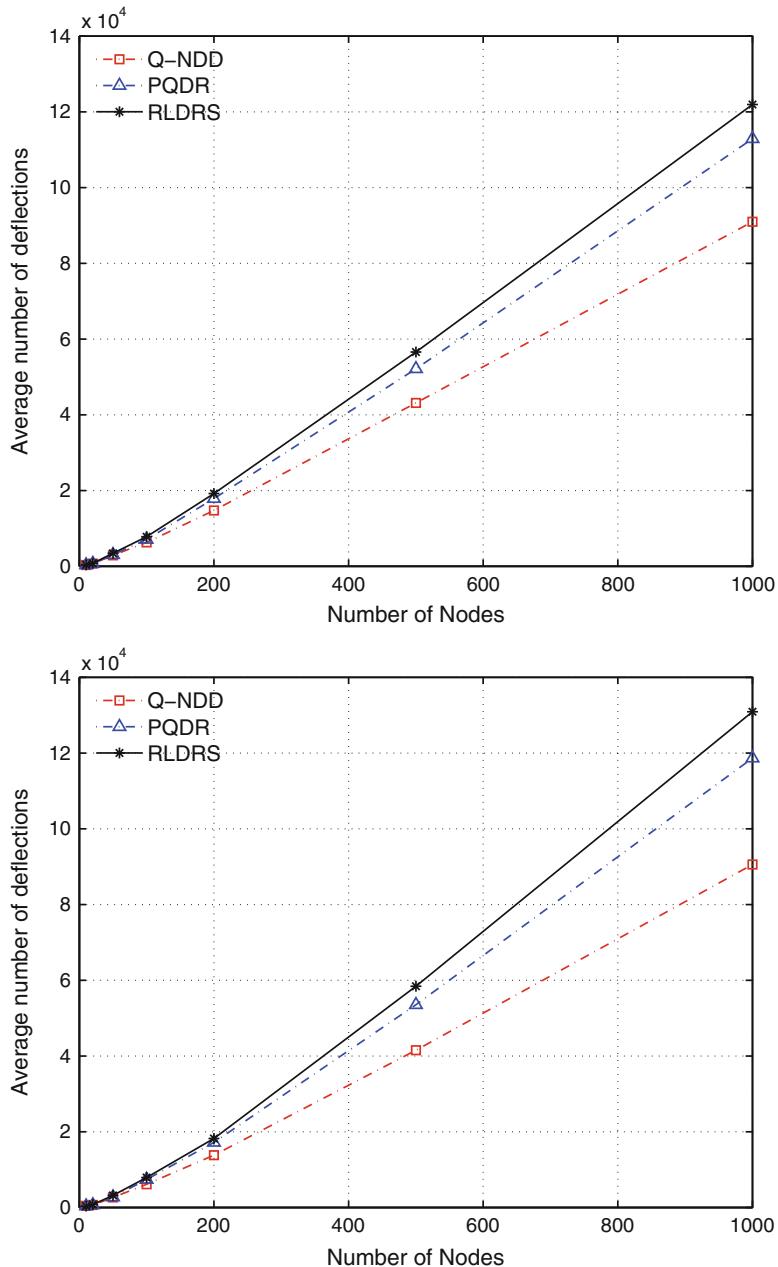


Fig. 15.11 Number of deflections as a function of the number of nodes in Waxman (top) and Barabási-Albert (bottom) networks

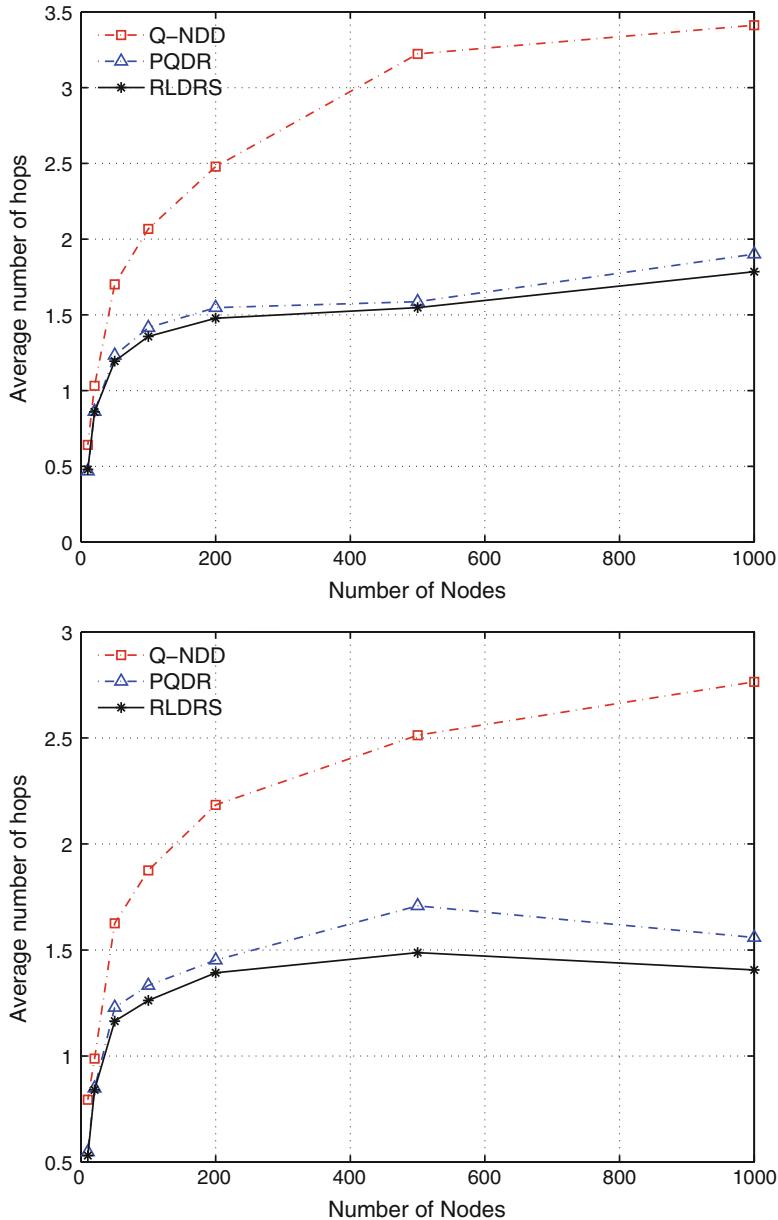


Fig. 15.12 Average number of hops traveled by bursts as a function of the number of nodes in Waxman (*top*) and Barabási-Albert (*bottom*) networks

15.5.4 Number of Deflections

Although burst deflection reduces the burst-loss probability, it introduces excess traffic load to the network. Number of deflections as a function of number of nodes for Waxman and Barabási-Albert network topologies is shown in Fig. 15.11. Q-NDD deflects fewer number of bursts compared to PQDR and RLDRS. Comparison of Waxman and Barabási-Albert network topologies shows an insignificant variation in the number of deflections.

15.5.5 Average Number of Hops

Average number of hops traveled by bursts as a function of the number of network nodes for Waxman and Barabási-Albert topologies is shown in Fig. 15.12. In case of Q-NDD, bursts travel through additional hops compared to PQDR and RLDRS. When deflecting a burst, PQDR and RLDRS consider the number of hops to destination. Simulation results show that the underlying topology and nodes connectivity have an impact on the number of hops traveled by bursts. Bursts travel fewer hops in case of Barabási-Albert networks. Consider the case of 1,000 nodes shown in Fig. 15.12. In the case of Waxman topology shown in Fig. 15.12 (top), the Q-NDD algorithm causes bursts to travel 3.5 hops on average while in the case of Barabási-Albert topology shown in Fig. 15.12 (bottom), number of traveled hops is only 2.75.

15.6 Conclusion

In this Chapter, we compared performance of the Q-learning-based Node Degree Dependent (Q-NDD) deflection routing algorithm, the Predictive Q-learning-based Deflection Routing (PQDR) algorithm, and the Reinforcement Learning Based Deflection Routing Scheme (RLDRS). Simulations were performed using complex network topologies that were generated by Waxman and Barabási-Albert algorithms.

The Q-NDD burst-loss probability is smaller and bursts are deflected less frequently than PQDR and RLDRS. However, bursts travel through additional hops and thus experience longer end-to-end delays. Therefore, smaller burst-loss probability and smaller number of deflections come at the cost of selecting longer paths, which are less likely to be congested. PQDR and RLDRS consider the number of hops to destination when deflecting bursts. This, in turn, causes the bursts to travel through shorter paths. However, the probability of congestion along shorter paths is usually higher because the majority of the routing protocols tend to route data through such paths. As a result, burst-loss probability and probability of defecting bursts is higher along the paths that PQDR and RLDRS select for deflection.

While the simulation results indicate that number of deflections does not significantly depend on the underlying topology, the bursts travel through fewer hops in Barabási-Albert networks. Q-NDD outperforms PQDR and RLDRS by exhibiting smaller burst-loss probability and smaller number of deflections. However, in the case of Q-NDD, the bursts travel through additional hops. The improved performance of Q-NDD becomes more evident as the size of the network increases.

References

1. Abd El-Rahman, A.I., Rabia, S.I., Shalaby, H.M.H.: MAC-layer performance enhancement using control packet buffering in optical burst-switched networks. *J. Lightw. Technol.* **30**(11), 1578–1586 (2012)
2. Acampora, A.S., Shah S.I.A.: Multihop lightwave networks: a comparison of store-and-forward and hot-potato routing. In: Proceedings of the IEEE INFOCOM, Bal Harbour, FL, USA, **1**, pp. 10–19 (1991)
3. Addie, R.G., Neame, T.D., Zukerman, M.: Performance evaluation of a queue fed by a Poisson Pareto burst process. *Comput. Netw.* **40**(3), 377–397 (2002)
4. Albert, R., Barabási, A.L.: Statistical mechanics of complex networks. *Rev. Mod. Phys.* **74**(1), 47–97 (2002)
5. Autonomous System Numbers (2014, Dec.) [Online]. Available: <http://www.iana.org/assignments/as-numbers/>
6. Baldine, I., Rouskas, G.N., Perros, H.G., Stevenson, D.: JumpStart: a just-in-time signaling architecture for WDM burst-switched networks. *IEEE Commun. Mag.* **40**(2), 82–89 (2002)
7. Barabási, A.L., Albert, R.: Emergence of scaling in random networks. *Science* **286**(5439), 509–512 (1999)
8. Baran, P.: On distributed communications networks. *IEEE Trans. Commun. Syst.* **CS-12**(1), 1–9 (1964)
9. Bathula, B.G., Vokkarane, V.M.: QoS-based multicasting over optical burst-switched (OBS) networks. *IEEE/ACM Trans. Netw.* **18**(1), 271–283 (2010)
10. Belbekhouche, A., Hafid, A., Tagmouti, M., Gendreau, M.: Topology-aware wavelength partitioning for DWDM OBS networks: a novel approach for absolute QoS provisioning. *Comput. Netw.* **54**(18), 3264–3279 (2010)
11. Belbekhouche, A., Hafid, A., Gendreau, M.: Novel reinforcement learning-based approaches to reduce loss probability in buffer-less OBS networks. *Comput. Netw.* **53**(12), 2091–2105 (2009)
12. Borgonovo, F.: Deflection routing. *Routing in Communications Networks*. Prentice-Hall, New Jersey, pp. 263–306 (1995)
13. Borgonovo, F., Fratta, L., Bannister, J.: Unslotted deflection routing in all-optical networks. In: Proceedings of the IEEE GLOBECOM, Houston, TX (1993)
14. Boyan, J.A., Littman, M.L.: Packet routing in dynamically changing networks: a reinforcement learning approach. In: Jack, J., Cowan, D., Tesauro, G., Alspector, J., (eds.) *Advances in Neural Information Processing Systems*, vol. 6, pp. 71–678. Morgan Kaufmann Publishers, San Francisco (1994)
15. Bregni, S., Caruso, A., Pattavina, A.: Buffering-deflection tradeoffs in optical burst switching. *Photon. Netw. Commun.* **20**(2), 193–200 (2010)
16. BRITE (2014, Dec.) [Online]. Available: <http://www.cs.bu.edu/brite>
17. Calvert, K.L., Dora, M.B., Zegura, E.W.: Modeling Internet topology. *IEEE Commun. Mag.* **35**(6), 160–163 (1997)
18. Chen, Q., Chang, H., Govindan, R., Jamin, S., Shenker, S., Willinger, W.: The origin of power laws in Internet topologies revisited. In: Proceedings of the INFOCOM, New York, USA, pp. 608–617 (2002)

19. Chen, Y., Qiao, C., Yu, X.: Optical burst switching: a new area in optical networking research. *IEEE Netw.* **18**(3), 16–23 (2004)
20. Chich, T., Cohen, J., Fraigniaud, P.: Unslotted deflection routing: a practical and efficient protocol for multihop optical networks. *IEEE/ACM Trans. Netw.* **9**(1), 47–59 (2001)
21. Choi, S.P.M., Yeung, D.Y.: Predictive Q-routing: a memory-based reinforcement learning approach to adaptive traffic control. In: Touretzky, D.S., Mozer, M.C., Hasselmo, M.E. (eds.) *Advances in Neural Information Processing Systems*, vol. 8, pp. 945–951. MIT Press, Cambridge (1996)
22. Erdős, P., Rényi, A.: On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci.* **5**, 17–61 (1960)
23. Faloutsos, M., Faloutsos, P., Faloutsos, C.: On power-law relationships of the Internet topology. *SIGCOMM Comput. Commun. Rev.* **29**(4), 251–262 (1999)
24. Gao, X., Bassiouni, M.: Improving fairness with novel adaptive routing in optical burst-switched networks. *J. Lightw. Technol.* **27**(20), 4480–4492 (2009)
25. GNU General Public License (2014, Dec.) [Online]. Available: <http://www.gnu.org/copyleft/gpl.html>
26. Greenberg, A., Hajek, B.: Deflection routing in hypercube networks. *IEEE Trans. Commun.* **40**(6), 1070–1081 (1992)
27. Haeri, S., Trajković, L.: Intelligent deflection routing in buffer-less networks. *IEEE Trans. Cybern.* **45**(2), 316–327 (2015)
28. Haeri, S., Trajković, L.: Deflection routing in complex networks. In: Proceedings of the IEEE International Symposium Circuits and Systems, Melbourne, Australia, pp. 2217–2220 (2014)
29. Haeri, S., Arianezhad, M., Trajković, L.: A predictive Q-learning-based algorithm for deflection routing in buffer-less networks. In: Proceedings of the IEEE International Conference Systems, Man, and Cybernetics, Manchester, UK, pp. 764–769 (2013)
30. Haeri, S., Thong, W.W.K., Chen, G., Trajković, L.: A reinforcement learning-based algorithm for deflection routing in optical burst-switched networks. In: Proceedings of the IEEE International Conference Information Reuse and Integration, San Francisco, USA, pp. 474–481 (2013)
31. Iyer, S., Bhattacharyya, S., Taft, N., Diot, C.: An approach to alleviate link overload as observed on an IP backbone. In: Proceedings of the IEEE INFOCOM, Stanford, CA, USA, vol. 1, pp. 406–416 (2003)
32. Izal, M., Aracil, J.: On the influence of self-similarity on optical burst switching traffic. In: Proceedings of the IEEE GLOBECOM, Taipei, Taiwan, vol. 3, pp. 2308–2312 (2002)
33. Kaelbling, L.P., Littman, M.L., Moore, A.W.: Reinforcement learning: a survey. *J. Artif. Intell. Res.* **4**, 237–285 (1996)
34. Kiran, Y., Venkatesh, T., Murthy, C.: A reinforcement learning framework for path selection and wavelength selection in optical burst switched networks. *IEEE J. Sel. Areas Commun.* **25**(9), 18–26 (2007)
35. Krakiwsky, E.J., Wells, D.E.: Coordinate systems in geodesy. In: Fredericton, NB: Lecture Notes LN# 16, Department of Geodesy and Geomatics Engineering, University of New Brunswick (1971)
36. Leland, W.E., Taqqu, M.S., Willinger, W., Wilson, D.V.: On the self-similar nature of Ethernet traffic (extended version). *IEEE/ACM Trans. Netw.* **2**(1), 1–15 (1994)
37. Levesque, M., Elbiaze, H., Aly, W.: Adaptive threshold-based decision for efficient hybrid deflection and retransmission scheme in OBS networks. In: Proceedings of the 13th International Conference Optical Network Design and Modeling, Braunschweig, Germany, pp. 55–60 (2009)
38. Li, S., Wang, M., Wong, E.W.M., Abramov, V., Zukerman, M.: Bounds of the overflow priority classification for blocking probability approximation in OBS networks. *J. Opt. Commun. Netw.* **5**(4), 378–393 (2013)
39. Liu, H.L., Zhang, B., Shi, S.L.: A novel contention resolution scheme of hybrid shared wavelength conversion for optical packet switching. *J. Lightw. Technol.* **30**(2), 222–228 (2012)

40. Lü, J., Chen, G., Ogorzalek, M., Trajković, L.: Theories and applications of complex networks: advances and challenges. In: Proceedings of the IEEE International Symposium Circuits and Systems, Beijing, pp. 2291–2294 (2013)
41. Maxemchuk, N.F.: Comparison of deflection and store and forward techniques in the Manhattan street and shuffle exchange networks. In: Proceedings of the IEEE INFOCOM, Ottawa, ON, Canada, vol. 3, pp. 800–809 (1989)
42. Merit/NSFNET Information Services: The technology timetable. *Link Lett.* **7**(1), 8–11 (1994)
43. Mountroudou, X., Perros, H.: On the departure process of the burst aggregation algorithms in optical burst switching. *J. Comput. Netw.* **53**(3), 247–264 (2009)
44. Najiminaini, M., Subedi, L., Trajković, Lj.: Analysis of Internet topologies: a historical view. In: Proceedings of the IEEE International Symposium Circuits and Systems, Taipei, Taiwan, pp. 1697–1700 (2009)
45. Nowe, A., Steenhaut, K., Fakir, M., Verbeeck, K.: Q-learning for adaptive load based routing. In: Proceedings of the IEEE International Conference System Man, and Cybernetics, San Diego, CA, USA, vol. 4, pp. 3965–3970 (1998)
46. Paxson, V., Floyd, S.: Wide-area traffic: the failure of Poisson modeling. *IEEE/ACM Trans. Netw.* **3**(3), 226–244 (1995)
47. Perelló, J., Agraz, F., Spadaro, S., Comellas, J., Junyent, G.: Using updated neighbor state information for efficient contention avoidance in OBS networks. *Comput. Commun.* **33**(1), 65–72 (2010)
48. Perros, H.G.: Connection-Oriented Networks: SONET/SDH, ATM, MPLS and Optical Networks. Wiley, Chichester (2005)
49. Peshkin, L., Savova, V.: Reinforcement learning for adaptive routing. In: Proceedings of the International Joint Conference Neural Network, Honolulu, HI, USA, vol. 2, pp. 1825–1830 (2002)
50. Qiao, C., Yoo, M.: Optical burst switching (OBS)—a new paradigm for an optical Internet. *J. High Speed Netw.* **8**(1), 69–84 (1999)
51. Siganos, G., Faloutsos, M., Faloutsos, P., Faloutsos, C.: Power-laws and the AS-level Internet topology. *IEEE/ACM Trans. Netw.* **11**(4), 514–524 (2003)
52. Subedi, L., Trajković, Lj.: Spectral analysis of Internet topology graphs. In: Proceedings of the IEEE International Symposium Circuits and Systems, Paris, France, pp. 1803–1806 (2010)
53. The Google Earth. (2014, Dec.) [Online]. Available: <http://www.google.com/earth/index.html>
54. A special report: a brief history of NSF and the Internet (2014, Dec.) [Online]. Available: http://www.nsf.gov/news/special_reports/cyber/internet.jsp
55. The ns-3 network simulator (2014, Dec.) [Online]. Available: <http://www.nsnam.org/>
56. Thong, W.W.K., Chen, G.: Jittering performance of random deflection routing in packet networks. *Commun. Nonlinear Sci. Numer. Simul.* **18**(3), 616–624 (2013)
57. Thong, W.W.K., Chen, G., Trajković, L.: RED-f routing protocol for complex networks. In: Proceedings of the IEEE International Symposium Circuits and Systems, Seoul, Korea, pp. 1644–1647 (2012)
58. Trajković, L.: Analysis of Internet topologies. *IEEE Circuits Syst. Mag.* **10**(3), 48–54, Third Quarter (2010)
59. Wang, W.X., Yin, C.Y., Yan, G., Wang, B.H.: Integrating local static and dynamic information for routing traffic. *Phys. Rev. E* **74**, 016101 (2006)
60. Wang, X., Jiang, X., Pattavina, A.: Efficient designs of optical LIFO buffer with switches and fiber delay lines. *IEEE Trans. Commun.* **59**(12), 3430–3439 (2011)
61. Watkins, C.J.C.H., Dayan, P.: Technical note. Q-learning. *Mach. Learn.* **8**(3), 279–292 (1992)
62. Watts, D.J., Strogatz, S.H.: Collective dynamics of small-world networks. *Nature* **393**, 440–442 (1998)
63. Waxman, B.M.: Routing of multipoint connections. *IEEE J. Sel. Areas Commun.* **6**(9), 1617–1622 (1988)
64. Wong, E.W.M., Baliga, J., Zukerman, M., Zalesky, A., Raskutti, G.: A new method for blocking probability evaluation in OBS/OPS networks with deflection routing. *J. Lightw. Technol.* **27**(23), 5335–5347 (2009)

65. Wu, G., Dai, W., Li, X., Chen, J.: A maximum-efficiency-first multi-path route selection strategy for optical burst switching networks. *Optik-Int. J. Light Electr. Optics* **125**(10), 2229–2233 (2014)
66. Xiong, Y., Vandenhouwe, M., Cankaya, H.C.: Control architecture in optical burst-switched WDM networks. *IEEE J. Sel. Areas Commun.* **18**(10), 1838–1851 (2000)
67. Xu, L., Perros, H.G., Rouskas, G.: Techniques for optical packet switching and optical burst switching. *IEEE Commun. Mag.* **39**(1), 136–142 (2001)
68. Yang, X., Wetherall, D.: Source selectable path diversity via routing deflections. In: *Proceedings of the ACM SIGCOMM*, New York, USA, pp. 159–170 (2006)
69. Yoo, M., Qiao, C., Dixit, S.: Comparative study of contention resolution policies in optical burst-switched WDM networks. In: *Proceedings of the SPIE*, Boston, MA, vol. 4213, pp. 124–135 (2000)
70. Yu, X., Li, J., Cao, X., Chen, Y., Qiao, C.: Traffic statistics and performance evaluation in optical burst switched networks. *J. Lightw. Technol.* **22**(12), 2722–2738 (2004)
71. Zalesky, A., Vu, H., Rosberg, Z., Wong, E.W.M., Zukerman, M.: Stabilizing deflection routing in optical burst switched networks. *IEEE J. Sel. Areas Commun.* **25**(6), 3–19 (2007)
72. Zalesky, A., Vu, H., Rosberg, Z., Wong, E., Zukerman, M.: OBS contention resolution performance. *Perform. Eval.* **64**(4), 357–373 (2007)
73. Zalesky, A., Vu, H., Rosberg, Z., Wong, E.W.M., Zukerman, M.: Modelling and performance evaluation of optical burst switched networks with deflection routing and wavelength reservation. In: *Proceedings of the INFOCOM*, Hong Kong SAR, China, vol. 3, pp. 1864–1871 (2004)
74. Zegura, E.W., Calvert, K.L., Donahoo, M.J.: A quantitative comparison of graph-based models for Internet topology. *IEEE/ACM Trans. Netw.* **5**(6), 770–783 (1997)
75. Zukerman, M., Neame, T.D., Addie, R.G.: Internet traffic modeling and future technology implications. In: *Proceedings of the IEEE INFOCOM*, San Francisco, CA, pp. 587–596 (2003)

Chapter 16

Recommender Systems for Social Networks Analysis and Mining: Precision Versus Diversity

Amin Javari, Malihe Izadi and Mahdi Jalili

Abstract Recommender systems has become increasingly important in online community for providing personalized services and products to users. Traditionally, performance of recommender algorithms has been evaluated based on accuracy and the focus of the research was on providing accurate recommendation lists. However, recently diversity and novelty of recommendation lists have been introduced as key issues in designing recommender systems. In general, novelty/diversity and accuracy do not go hand in hand. Therefore, designing models answering novelty/diversity-accuracy dilemma is one of the challenging problems in the context of practical recommender systems. In this paper, we first introduce the diversity-accuracy challenge in recommender systems, and then present two recommendation algorithms which approach the problem from two perspectives. The first model is a filtering algorithm to select candidate items which incorporates timing information of ratings to improve both accuracy and novelty of recommender systems. The filter can be applied as adds-on to any recommender algorithm. The second model is a probabilistic model which resolves the dilemma and provides adjustable level of accuracy and diversity that can be tuned by a single parameter.

16.1 Introduction

Due to the increasing volume of available data on the internet, in many e-commerce systems, users encounter millions of options and items which they may purchase. Exploring such large item space and choosing the best matches for the users inter-

A. Javari · M. Izadi

Department of Computer Engineering, Sharif University of Technology, Tehran, Iran
e-mail: a.javari@gmail.com

M. Izadi

e-mail: malihe.ie69@gmail.com

M. Jalili (✉)

School of Electrical and Computer Engineering, RMIT University, Melbourne, VIC 3001,
Australia
e-mail: mahdi.jalili@rmit.edu.au

ests, is very time consuming and in many situations is not manageable for them. Concerning this issue, recommender systems have become a core component for many e-commerce applications to help users find what they are looking for. Recommender systems can be considered as information retrieval systems, in which there is no direct request for information. Indeed, recommender systems have two major tasks. First to extract users interests based on various information resources, and then to provide a list of items which maximizes their satisfaction. Based on the type of information used for the recommendation, they can be generally categorized into three main classes: content-based, collaborative filtering and hybrid algorithms [1]. A content-based recommender first extracts a profile for each user based on the content of the items accessed by that user. Then, the system recommends items to the target user which has the most similar content to the users profile. In collaborative filtering algorithms, system is blind to the contents of items and uses the previous user-item interactions. Algorithms based on collaborative filtering make recommendations for any user by the help of collective preferences of other users [2]. Hybrid methods use both types of the information resources [3]. In many applications, content of items is not easy to extract, and collaborative filtering models are the only choice.

In a classic manner of recommender systems, the recommendation list is provided such that the target user likely gives high ratings to the recommended items. However, recently it has been shown that, a recommender system should not only have a good accuracy but also its generated recommendation lists should have proper novelty and diversity [4, 5]. In general, novelty and diversity of recommendations represents the ability of recommender models to suggest items which the users would not discover them by themselves. Diversity of recommendations has two forms: diversity of a single recommendation lists called intra-list diversity and diversity of a set of recommendation lists known as inter-list diversity. Intra-list diversity describes that how much the contents of items in a recommendation list are similar to each other. While, inter-list diversity explains the similarity of the list recommended to a set of users [6]. Apparently, as the similarity between the recommendation lists is lower, the system provides more diverse and personalized recommendations. In this paper, our focus is on inter-list diversity. Diverse recommendations can be achieved by giving the same chance of recommendation to the whole items set without any focus on items with specific popularity interval. However, higher accuracy can be obtained by giving more weight on recommending popular items. In other words maximizing accuracy and diversity of the recommendation list are not in the same direction [7].

Novelty of recommendations is another issue that should be considered in the recommendation process. Novelty, describes that how much recommendations are informative for users [4]. Most of the algorithms based on crowd preferences, which are minimizing the risk of recommendation in terms of accuracy, are biased towards popular items [8]. Apparently, there is a dilemma between accuracy and diversity/novelty in recommendations. In recent years, solving this dilemma has received attention and some algorithms have been proposed to provide recommendations supporting novelty and diversity [5, 7, 9, 10]. Zhou et al. (2010) introduced a hybrid algorithm taking into account accuracy and diversity of recommendation and

demonstrated its functionality in solving the accuracy-diversity dilemma of recommender systems [7]. Adomavicious et al. (2011) employed graph theoretic methods based on maximum bipartite matching to maximize recommendation diversities [11]. In the same direction, recently we have studied the diversity, novelty and accuracy dilemma in three different works. We have shown that by utilizing contextual information, in particular time label of ratings, can improve both novelty and accuracy of recommendations [12]. In another work, we introduced a probabilistic model which solves accuracy-diversity dilemma over Markov-based models [13]. And finally, we proposed a novel recommendation model based on extracting cascade networks among items [14]. We have shown that by relying on extracting cascade network among items, it is possible to detect novel and relevant items for the target users profile. In the rest of the paper, we first explain the diversity/novelty-accuracy dilemma and introduce some metrics for evaluating recommender systems. Then, we illustrate our two proposed models answering the problem.

16.2 Datasets

The benchmark datasets used here are MovieLens and Netflix. MovieLens is probably the most popular movie-based dataset used for this purpose. The ratings are explicit and between 1 to 5 stars. Similarly, Netflix consists of ratings of different movies, explicit and in the range of 1 to 5 stars. The ratings matrix can be converted to a binary form in which 1 means the user likes the item and 0 means she probably dislike it. The conversion can be done based a threshold (e.g. the average rating of each user or the total average rating). The experiments here are based on subsets of the datasets.

16.3 Diversity and Novelty

Although there are various evaluation metrics for assessing the performance of recommender systems, it is not easy to choose the best suitable measure as well. On one hand, there are accuracy-based metrics, and on the other hand there are diversity, novelty and coverage measures. Despite the fact that classic recommenders generate fairly enough accurate recommendations, it has been shown that recommending items solely based on maximizing the accuracy may result in recommending only a small portion of the item space. Moreover, recommendation lists will be biased towards items with high popularity. It has been showed by various studies that there exists an inconsistency between the values of these metrics in different recommender algorithms [4, 5, 7]. More specifically, consider the degree distribution of available items in a system. In order to increase the precision of recommendations, items with higher degrees need to have a higher chance of being recommended. The reason is, according to the maximum likelihood, items with higher popularity in a specific

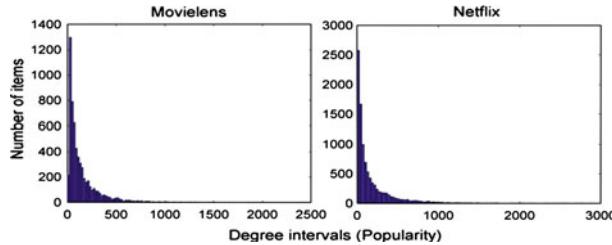


Fig. 16.1 Degree distribution of items in Netflix and Movielens datasets

time interval are more probable to get accessed in future time steps. On the other hand, items with lower degrees increase the novelty. Finally, to improve coverage and diversity of recommendations, we need to suggest items with different popularity degrees (i.e. from the whole range). For example, consider a recommender which recommends popular items to users, the precision of such a system is normally high. However, the novelty is low because most of the recommended items are already known or even used by users. Furthermore, diversity and coverage are low as well, because there are a limited number of popular items recommended to everybody.

To better illustrate this inconsistency, consider Fig. 16.1 which presents the degree distribution of the two datasets. In both datasets about 20 % of the items attract 80 % of users ratings, thus a large number of items in the item space have low degrees, and consequently low popularity values. When we focus on items with low degree, we can recommend many of them. However, recommending items with higher values of degrees (i.e. increasing accuracy) forces the model to lose coverage and diversity. On the other hand, diversity insist on always recommending items from the whole range of item set, while novelty increases when we recommend mostly low degree items.

16.3.1 Evaluation of Recommendation Lists

There are two approaches for evaluation of recommenders, offline or online [15]; in an offline evaluation, we first collect the dataset, then extract a proportion of ratings as a test set, and name the rest as the training set, which will be used for training the recommender algorithm and finally the recommendation lists are evaluated using a set of metric. There are various number of evaluation metrics for evaluating accuracy, diversity and novelty [4, 16]. Some of them are introduced in the following.

Precision. Precision is an accuracy-based metric for evaluation of the top-N recommendation task. $P_u(N)$, the precision of the list recommended to user u , is defined as the number of relevant items to this user which are present in the recommendation list. N is the size of recommendation list and relevant items, are those which the

target user has liked in the test set. Precision is computed using

$$P_u(N) = \frac{TP_u}{N}, \quad (16.1)$$

where TP_u is the number of relevant items recommended by system to user u . The average precision of the system, $P(N)$, is the mean precision of the lists recommended to all test users.

Diversity. Recommendation lists need to contain more diverse and personalized items according to the preference of different users. Inter-list diversity deals with the personalization issue in those lists. $D(N)$ can be defined as the average distance of recommendation lists for all test users by a recommender algorithm. The more distance between lists, the more personalized ones we have. The distance of two lists recommended to users i and j is

$$d_{i,j} = 1 - \frac{c_{i,j}}{N}, \quad (16.2)$$

where $c_{i,j}$ is the number of common items in their recommendation lists.

Coverage. Diversity measures how similar are the recommendation lists of different users. We can apply Shannon entropy theory to add the concept of item coverage to diversity. Entropy-based coverage measures what percentage of items are recommended and how evenly they are distributed in the recommendations lists of users using the following equation

$$EC = - \sum_{i=1}^n p_i \log_2 p_i, \quad (16.3)$$

where p_i is the percentage of the recommendation lists that contain item i and n is the total number of items.

Novelty. Good recommenders try to provide users with new information about novel items and suggest fairly non-obvious items. To this end, some metrics have been introduced to evaluate novelty of recommendations lists. According to self-information based novelty (SIBN), users are more likely to know items which are popular. Therefore, $SIBN_u(N)$, will be measured for a list of recommendations for the user u , as

$$SIBN_u = \log \left(\frac{|U|}{d_i} \right), \quad (16.4)$$

where $|U|$ is the number of users in training set and d_i represents the number of users that have given ratings to item i . There is a modified version of this metric which considers the relevancy of items for the target user while calculating the novelty [13].

It is logical to consider only the novelty of items that the target user likes (according to his or her ratings). Effective *SIBN* is defined as

$$ESIBN = \frac{1}{|U|} \sum_{i=1}^N R_i \times SIBN_u. \quad (16.5)$$

where R_i is the relevancy of item i for the target user. This means if she or he likes it, it equals to 1, otherwise it is zero. Finally, the overall novelty of an algorithm is the average novelty of the recommendations to all test users.

Unifying inconsistent evaluation metrics in recommender systems. Considering the variety in evaluation metrics, assessing the performance of recommenders is not a trivial task. As showed by different studies, there is a tradeoff between the values of above measures. For instance, as the accuracy of a recommender increases, its novelty and diversity often decrease and vice versa. In other words, we can rarely find two recommenders in which the precision and diversity of first algorithm would be higher than the second at the same time. This issue makes it difficult to fairly compare different recommendation algorithms based on different metrics. In a recent work, we proposed a new evaluation model which answers the problem and integrates these metrics together in order to the performance of recommender algorithms more properly [17]. The fundamental principle in this evaluation model is to fix the value of one metric (e.g. precision) and compute the maximum possible bounds of others (e.g. novelty and diversity) for this value in a dataset.

16.4 Method

In the previous sections, we reviewed different metrics for evaluation of recommender systems and introduced an integrated framework for proper evaluation of recommenders. In this section, we first review some base-line recommender algorithms, and then introduce two methods to address this inconsistency from two other perspectives.

16.4.1 Baseline Algorithms

Collaborative Filtering (CF) algorithms recommend items based on the similarity between users or items [18]. More specifically, if users u and u' have similar tastes in part of their common rated items, their ratings of uncommon items impact the future recommendations for each one. CF methods are generally considered in two groups of memory-based and model-based algorithms. Memory-based CF methods use the entire ratings of users in order to make the recommendations. User-based CF and item-based CF are two important algorithms of this group. Item-based CF computes

the similarity of items based on their ratings, then using this similarity; it will predict a rating for an item-user pair. Due to their simple implementation they have been used in many commercial systems, but they have some drawbacks such as sparsity, cold start and scalability issues. On the other hand, model-based recommenders use statistical and machine learning techniques in order to learn a model based on the ratings matrix. This group includes algorithms such as Markov-based models [19], latent semantic and matrix factorization methods [20]. Various studies have applied Markov models for recommender systems. For instance, in order to model the recommendation problem, Shani et. al. used Markov decision processes (MDP) and defined a utility function for item set as a sequential optimization problem [19]. Moreover, Garcin et. al. proposed a model based on evolving context trees as variable order Markov models [21].

16.4.2 Novel and Accurate Recommendations Based on Popularity Forecasting

This algorithm generates novel and accurate recommendations based on analyzing item popularity time series. We have used wavelet transform to analyze popularity time series and predict their behavior in the future [22]. Based on analyzing popularity, a filtering algorithm for selecting candidate items was introduced which is called Novelty and popularity based filtering algorithm. The model prefers items with low popularity in the past which are going to be popular in the future based on our predictions. At first, the filtering algorithm selects a subset of items based on analyzing popularity time series of items. Afterward, any recommender can be applied to suggest personalized list of items to the target user from the selected subset of items. We used item-based CF and Markov-based recommender algorithm as baseline algorithms in the experiments. Our results indicate that the filter can significantly improve both the accuracy and effective novelty of the two mentioned recommenders.

More specifically, this filter incorporates contextual information to answer the accuracy-novelty dilemma. Contextual information describes the context in which users interact with the system [23]. Among different resources for contextual information, time is one of the most informative entities which are usually collected along the ratings. It is logical that users preferences change over time and time-aware recommenders use this fact to increase the quality of recommendations [24]. Although these models have enhanced the precision of recommendations, they do not focus on novelty and diversity of the lists. Therefore in our model, dynamics of item popularity in the time-domain were exploited to generate both novel and accurate recommendations. We showed that in each time step, there is a subset of items which are not popular, but will be probably preferred by users in near future. Figure 16.2 sketches daily popularity values for a sample movie. According to our findings usually items that have received many ratings, show an increased popularity in a certain time interval after which they become unpopular gradually.

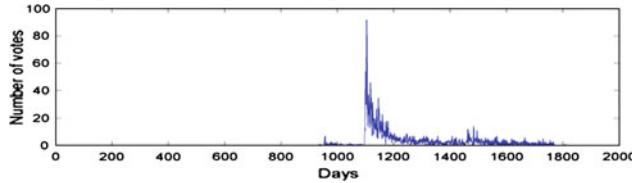
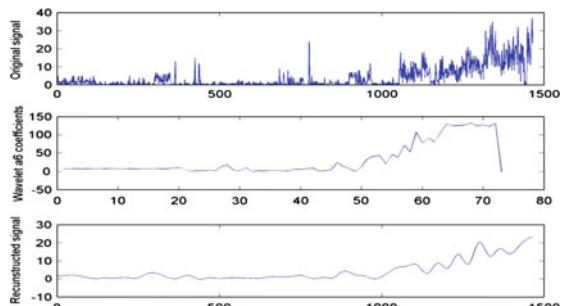


Fig. 16.2 Popularity (number of votes) time series for a sample item from Netflix dataset

Our model attends to unpopular items that are expected to be popular in near future. Since such items are currently non-popular, they are novel for users. Also, they will be popular in the future, thus they are accurate as well. In the following, we first explain how to predict these popularity trends using the wavelet decomposition method and show how to use them to select the top-N items and finally present our results.

Predicting popularity time series based on wavelet decomposition. We use a method based on wavelet transform to analyze and predict popularity time series of items [22]. Time series prediction based on wavelet transforms has three steps. First, wavelet coefficients in multiple levels are computed using wavelet decomposition. It has been proved that the signals obtained from decomposition are more predictable (have lower complexity) than the original signal. Then, the signals go through signal extinction. In the end, the coefficients obtained in the signal extinction step are used in wavelet reconstruction stage. It is worth mentioning that an N -level decomposition of a time series produces N distinct time series (a_1, \dots, a_N) , where a_N and a_1 are respectively the highest and the lowest dynamics of a . The number of needed decomposition levels is based on signals complexity. Here, we first applied different number of levels to find the level in which the signal describing the lowest dynamics of the original time series does not have high frequency patterns. We only want to predict the future popularity trends, not the exact values, thus we decompose in 6 levels and perform the extinction on the a_6 signal. Upper panel of Fig. 16.3, presents the popularity time series of a sample item, while the middle panel depicts the a_6 signal obtained from decomposition in 6 levels. Through performing extinction on

Fig. 16.3 Popularity time series for a sample item (upper panel), decomposed wavelet coefficient at level 6 (middle panel), and reconstructed signal from this coefficient (bottom panel)



a_6 and reconstructing from this signal, we will have the main trend prediction of the input signal (bottom panel of Fig. 16.3). In the next section, we explain how to select the top-N items using predicted values of the popularity time series.

Item selection for top-N recommendation based on information extracted from popularity time series. We introduce a filtering algorithm for selection of candidate items based on the popularity trends over time, *Popularity and Novelty Score* (PNS) based filtering. The main strategy of the PNS filter is to collect items with low popularity values in the past that display increasing popularity in the future. To this end, first we define a PNS value for each item, and then select top M items which have the highest PNS value. PNS value is defined as

$$PNS_i(t) = APP_i(t, t + m) \times NS_i(t) \quad (16.6)$$

where $APP_i(t, t + m)$ represents the average predicted popularity value of item i in m future time steps and Novelty Score (NS) of item i is calculated by

$$NS_i(t) = \frac{\log\left(\frac{HP(t)}{NV_i(t)}\right)}{\log\left(\frac{HP(t)}{LP(t)}\right)}, \quad (16.7)$$

where $NV_i(t)$ is the total number of ratings for item i up to time step t . Furthermore, $HP(t)$ indicates the total number of ratings for the most popular item up to time t and LP is the total number of ratings for the least popular item up to time t . By increasing novelty and popularity in future, PNS value will also increase as well. It is worth mentioning that the NS value defined above, have the same concept as SIBN; since if an item is more novel, it will have higher NS value. After calculation of PNS value for all items, PNS filter chooses the top-M items as candidates with the highest PNS value.

Comparing the proposed algorithms with baseline recommendation algorithms. The performance of the proposed model is compared with three other algorithms including item-based CF, Markov recommender and time decayed item-based CF [25]. The algorithms are evaluated on a subset of Netflix dataset with 60000 users, 4000 items and 6600000 ratings from users to items. Moreover, the dataset includes timings information at which the ratings have been made. In our evaluations, we first select five test points uniformly distributed on the data time interval and, then, take the ratings up to the that time as training data and those after that point up to the next 20 days as test data. This results in five different train and test datasets. Using train data, we recommend each test user a list of recommendations with 10 items and evaluate it based on test data. A parameter which largely influences performance of the PNS-filter is the basket size. By decreasing the basket size, the filtering algorithm will be more influential on recommendations. However, if we use small number of candidate items, the recommendation algorithm in the second step will lose its proper functionality. In general, the optimal basket size can be determined based on factors like size of the item space, the number of items in recommendation lists and

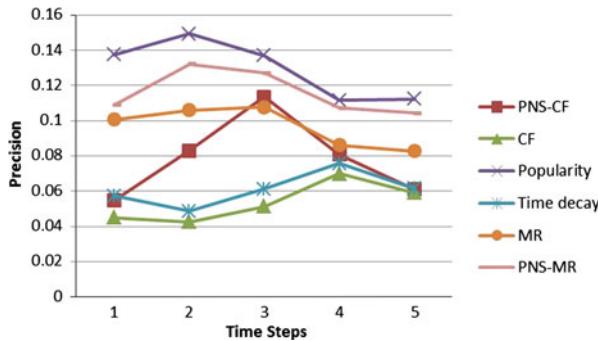


Fig. 16.4 Precision of six recommendation algorithms (PNS-CF, PNS-MR, Collaborative Filtering (CF), Popularity-based recommender (Popularity), Markov Recommender (MR) and time-decay collaborative filtering (Time Decay)) on five time steps of Netflix dataset

distribution of items predicted popularity. In these experiments, we do not optimize the basket size and fix it as 50.

Figure 16.4 shows the precision of six recommendation algorithms in five test points. As it is seen, PNS-CF and PNS-MR show better precision as compared to CF and MR, respectively. It is worth mentioning that classical item-based CF and time decay algorithm as one of the time-aware recommendation algorithms C have almost the same precision. Also, we compare novelty of the recommendation list, as denoted by ESIBN measure and the results are shown in Fig. 16.5. It is seen that PNS-based recommendation methods have the highest ESIBN in many of the test points. As we discussed, PNS filter aims at maximizing the precision and besides takes into account novelty of recommendations. Thus, we expect higher ESIBN and precision for PNS-based methods compared to classic baseline ones: CF and MR.

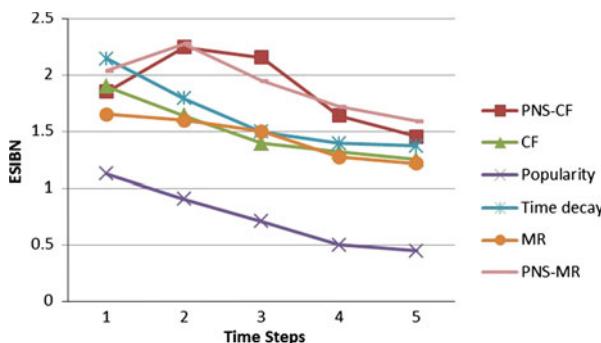


Fig. 16.5 Effective self-information based novelty (ESIBN) of eight recommendation algorithms. Other designations are as Fig. 16.1

16.4.3 Diversity and Novelty Dilemma in Markov-Based Recruiters

As mentioned, previous studies have not been able to balance between diversity and accuracy of recommendations. In this work, we propose a new probabilistic model that tackles some of the defects of classical Markov Chain (MC)-based algorithms and introduce a structure for solving the dilemma between accuracy and diversity by introducing a hybrid model, which balances two recommenders; one specialized for high accuracy and the other one for high novelty [13]. This structure can be applied to all MC-based recommenders without adding to the computational complexity of these models and can provide a flexible and tunable level of diversity and precision. More details on the proposed model are presented in the following sections.

State automata-based recommender system. Our model uses the sequential manner of users' ratings. Suppose each item is a state; therefore, one can model users sequential ratings as transition between different states. We consider two states for each item, to be able to add the value of item ratings to the transition probability; positive (like) state and negative (dislike) state which we referred to as L-state and D-state from now on. L-states and D-states of item i are denoted by $s_{i,L}$ and $s_{i,D}$, respectively. In fact, items will be modeled with $2I$ (I is number of items) states while users ratings are the transitions between these states. In other words, we model consecutive ratings of a user who gives positive rating on i and then gives negative rating on j as an edge $(s_{i,L}, s_{j,D})$. As an example, four ratings of a user presented in table 1 are transformed into the state transition model in Fig. 16.6 as an example.

Based on this state space, we introduce a graph to gather information about the users sequential behavior which is called as aggregated transition graph (AT) that has one node for each state of the introduced state space in which $n_{i,L}$ represents the state $s_{i,L}$. Also, the weight of the edge from $n_{i,L}$ to $n_{j,L}$, is defined as the number of users who rated item j after item i and have positive rating toward these items. In the next section, we explain our recommendation models using this graph. Here, we propose two probabilistic recommendation models based on the AT graph; one aims at increasing precisions, while the other one tries to recommend items that match the target users taste in the best way.

Recommendation based on precision maximization. In order to generate recommendations, first we calculate the probabilities of users interest in each item, and then, suggest items with the highest probability values. To this end, state of each user

Table 1. Rating history of a specific user

Item ID	Rate	Time Step
23	Dislike	3
532	Like	23
43	Like	48

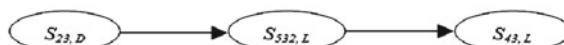


Fig. 16.6 The state transition model of the ratings history in table 1

is defined as a vector including her previous ratings of items. For instance, state of the user in Fig. 16.6 is defined as $< 23D, 532L, 43L >$. Accordingly $P(I_i|S_u)$, the probability of the interest of user u in item i , is computed by the following equation,

$$P(I_i|S_u) = P(s_{i,L}|S_u) - P(s_{i,D}|S_u), \quad (16.8)$$

where S_u represents the state of the user u , $P(I_i|S_u)$, the probability of transition from the state of user u to L-state of item i and finally $P(s_i, D|S_u)$, the probability of transition to D-state of the item i . In other words, recommendation score of item i is equal to the probability that user u likes this item minus the probability that it dislikes the item. Computing the transition probability of user u to the sub-state $s_{i,L}$ has 2 steps; first we need to calculate the probability of transition from sub-states visited by he/she to $s_{i,L}$, then unit these values in a weighted state. Probability of transition between sub-states is computed using users behavior from the training set. Therefore, the probability of the interest of user u in item i is

$$P(s_{i,L}|S_u) = \sum_{k=0}^m P(s_{i,L}|S_u(k)) \times \frac{1}{m}, \quad (16.9)$$

where m is the size of the state vector for user u and the probability of transition from sub-state k of user u to $s_{i,L}$ is denoted by $P(s_{i,L}|S_u(k))$, which can be computed using maximum likelihood estimation method. Therefore, the transition probability between node $n_{i,x}$ and $n_{j,y}$ is the weight of the edge $(n_{i,x}, n_{j,y})$ in the AT graph divided by out-degree of node $n_{i,x}$. $P(s_{i,D}|S_u(k))$ is calculated similarly.

After computing these probabilities for all items, we will recommend the top-N items with the highest predicted values for each user. Despite its good precision, this model which, denoted by Precision Maximizer (PM), is biased towards popular items and suffers from low novelty and diversity. Therefore, in the next section we propose the second model which will improve novelty and diversity of the recommendations. Recommendation based on specification maximization. Specification Maximizer (SM) model as its name suggests, tries to maximize user satisfaction through finding a list of items that is specific to the target user. The first step is to compute probabilities of specification for all candidate items regarding the target user and then suggest items with the highest probability values. The probability of specification is computed as

$$P(S_u|I_i) = P(S_u|s_{i,L}) - P(S_u|s_{i,D}), \quad (16.10)$$

where $P(S_u|I_i)$ indicates that if we suggest item I_i to user u , how specific is interest of user u in item I_i and it can be computed using two other probabilities; $P(S_u|s_{i,L})$ and $P(S_u|s_{i,D})$. $P(S_u|s_{i,L})$ can be obtained as

$$P(S_u|s_{i,L}) = \sum_{k=0}^m P(S_u(k)|s_{i,L}) \times \frac{1}{m}, \quad (16.11)$$

where $P(S_u(k)|s_{i,L})$ represents the probability of conditional transition from $S_u(k)$ to $s_{i,L}$. $P(S_u(k)|s_{i,L})$ indicates that if a user has visited $s_{i,L}$, in which probability the former sub-state visited by the user is $S_u(k)$. Therefore, when an item is relevant to a user and irrelevant for the others, it will more probably get recommended to this user. Since generally low degree items are more specific, our model will recommend these items more. Therefore, it will provide higher novelty. Hybrid recommender. The hybrid model linearly aggregates the outputs of SM and PM models. The probability vector of the proposed hybrid model, PCP, is computed as

$$P_{CP} = \alpha P_{SM} + (1 - \alpha) P_{PM}, \quad (16.12)$$

where PSM and PPM are respectively the probability vectors calculated by PM and SM models. Parameter can be tuned in the range of [0, 1] depending on the quality needed by the system owner. As α increases, novelty enhances, whereas small value of focuses on more precision. SM and PM models could be combined in other ways. For instance, items can be ranked based on the values of PSM and PPM, generating a ranking vector extracted from each of the models (called RPM and RSM). Then, the vector of the rankings can be linearly aggregated as the following

$$R_{CP} = \alpha R_{SM} + (1 - \alpha) R_{PM}, \quad (16.13)$$

where the extracted vector, RCP, can be used to generate the final recommendations.

Evaluation of the proposed model. In order to evaluate performance of the proposed recommendation model with that of three other classic algorithms classic Markov model, memory-based CF (item-based CF and user-based CF) and popularity-based recommender, we applied them on the datasets. In order to assess this feature of the model, we evaluated performance of the hybrid model as a function of hybridization parameter α . Figure 16.7 shows performance of the proposed method in terms of different metrics. As expected, PM model (the value of $\alpha = 0$)

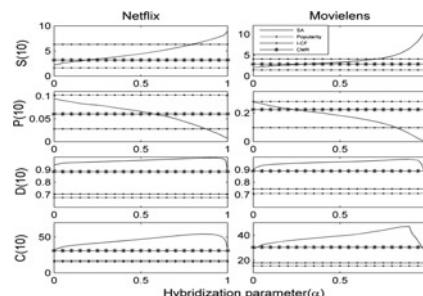


Fig. 16.7 Performance of the proposed recommendation method, item-based CF (I-CF), popularity-based recommender (Popularity) and classic Markov recommender (CMR) on Netflix and MovieLens datasets as a function of hybridization parameter. The methods are evaluated based on novelty $S(10)$, precision $P(10)$, diversity $D(10)$, and coverage $C(10)$

in both datasets results in the highest precision and the lowest novelty, whereas SM model ($\alpha = 1$) provides the highest novelty and the lowest precision. Moreover, as it can be seen, in order to achieve the desired performance, α can be tuned so that the target values for diversity and accuracy are obtained. Indeed, SM favors items with low degree, while PM is biased toward items with high degree. In other words, each of PM and SM models has focus on only particular part of item space and their recommendations partially cover the item space. This restricted focus on items with specific popularity causes these models to have low diversity. However, our results show that, diversity measures of the hybrid model reached its maximum for $\alpha \sim 0.9$ in both datasets. Indeed, by combining these two models, the hybrid model overcomes this restriction and makes it possible to include whole item space in recommendations. Moreover, according to our experiments, the proposed hybrid model outperforms the classic recommenders in terms of accuracy, novelty and diversity. We could always find a value of such that both precision, diversity and novelty of the proposed recommendation algorithm outperforms classic Markov model, item-based CF and popularity-based methods.

16.5 Conclusion

Recommender systems are widely used in many systems such as online social networks and music distributions and help users to easily find what they look for. Traditionally the most important issue addressed in the context of recommender systems is to support precision of the recommendation list, i.e., providing a recommendation list which maximizes the accuracy on the test dataset. By emerging new evaluation metrics such as diversity and novelty, and the significant inconsistency between these metrics and accuracy, the need to design recommendation algorithms which consider all the metrics has grown. Therefore, diversity and novelty of recommendations recently has become a challenging problem in this field. In this paper, we first investigated the diversity-accuracy dilemma by analyzing degree distribution of item space. Then, we introduced two algorithms concerning the challenge. First, a filtering algorithm for selecting candidate items based on items popularity. In this model, using wavelet decomposition method, slow dynamics of items popularity time series were forecasted. Based on these values, the model filters items from recommendation which have high popularity in the past time that show decreasing popularity. Our results showed that the model can help in enhancing novelty and accuracy of recommendations. Moreover, we introduced a Markov based recommender, which has an adjustable level of precision and diversity that can be tuned by a single parameter. The experiments showed that the model can successfully deal with the challenge of novelty-precision in recommendation systems.

References

1. Ricci, F., Shapira, B.: *Recommender Systems Handbook*. Springer, Heidelberg (2011)
2. Sarwar, B., et al.: Application of dimensionality reduction in recommender system-a case study. DTIC Document (2000)
3. Ricci, F., Rokach, L., Shapira, B.: Introduction to recommender systems handbook. *Recommender Systems Handbook*, pp. 1–35. Springer, Berlin (2011)
4. Vargas, S.: Novelty and diversity enhancement and evaluation in recommender systems and information retrieval. In: Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval, ACM (2014)
5. Santos, R.L., et al.: Diversity and novelty in information retrieval. In: SIGIR (2013)
6. Vargas, S., Castells, P.: Rank and relevance in novelty and diversity metrics for recommender systems. *Recommendation Systems*. Chicago, ACM (2011)
7. Zhou, T., et al.: Solving the apparent diversity-accuracy dilemma of recommender systems. *Proc. Natl. Acad. Sci. USA* **107**(10), 4511–4515 (2010)
8. McNee, S.M., Riedl, J., Konstan, J.A.: Being accurate is not enough: how accuracy metrics have hurt recommender systems. In: CHI’06 Extended Abstracts on Human Factors in Computing Systems, ACM (2006)
9. Adomavicius, G., Kwon, Y.: Maximizing aggregate recommendation diversity: a graph-theoretic approach. In: Proceedings of Workshop on Novelty and Diversity in Recommender Systems (2011)
10. Liu, J.G., Shi, K., Guo, Q.: Solving the accuracy-diversity dilemma via directed random walks. *Phys. Rev. E* **85**(1), 016118 (2012)
11. Adomavicius, G., Kwon, Y.: Improving aggregate recommendation diversity using rank-ing-based techniques. *IEEE Trans. Knowl. Data Eng.* **24**(5), 896–911 (2012)
12. Javari, A., Jalili, M.: Accurate and novel recommendations: an algorithm based on popularity forecasting. *ACM Trans. Intell. Syst. Technol. (TIST)* **5**(4), 56 (2014)
13. Javari, A., Jalili, M.: A probabilistic model to resolve diversity-accuracy challenge of recommendation systems. *Knowl. Inf. Syst.* 1–19 (2014)
14. Daneshmand, S.M., et al.: A time-aware recommender system based on dependency network of items. *Comput. J.* 115 (2014)
15. Beel, J., et al.: Research paper recommender system evaluation: a quantitative literature survey. In: Proceedings of the International Workshop on Reproducibility and Replication in Recommender Systems Evaluation, ACM (2013)
16. Gunawardana, A., Shani, G.: A survey of accuracy evaluation metrics of recommendation tasks. *J. Mach. Learn. Res.* **10**, 2935–2962 (2009)
17. Izadi, M., Javari, A., Jalili, M.: Unifying inconsistent evaluation metrics in recommender systems. RecSys conference, REDD workshop (2014)
18. Ekstrand, M.D., Riedl, J.T., Konstan, J.A.: Collaborative filtering recommender systems. *Found. Trends Hum. Comput. Interact.* **4**(2), 81–173 (2011)
19. Shani, G., Brafman, R.I., Heckerman D.: An MDP-based recommender system. In: Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann Publishers Inc. (2002)
20. Hofmann, T.: Latent semantic models for collaborative filtering. *ACM Trans. Inf. Syst. (TOIS)* **22**(1), 89–115 (2004)
21. Garcin, F., Dimitrakakis, C., Faltings, B.: Personalized news recommendation with context trees. (2013)
22. Soltani, S.: On the use of the wavelet decomposition for time series prediction. *Neurocomputing* **48**(1–4), 267–277 (2002)
23. Adomavicius, G., Tuzhilin, A.: Context-aware recommender systems. *Recommender Systems Handbook*, pp. 217–253. Springer, Berlin (2011)
24. Xiang, L., Yang Q.: Time-dependent models in collaborative filtering based recommender system. In: International Joint Conference on Web Intelligence and Intelligent Agent Technology. IEEE, Washington (2009)

25. Ding, Y., Li, X.: Time weight collaborative filtering. In: International Conference on Information and Knowledge Management. ACM, Germany (2005)

Chapter 17

Strategy Selection in Networked Evolutionary Games: Structural Effect and the Evolution of Cooperation

Shaolin Tan and Jinhua Lü

Abstract Networked evolutionary games provide an appropriate tool for investigating competition and diffusion of behavioral traits in structured biological and social populations. A core challenge in networked evolutionary game theory is the strategy selection problem: Given several strategies, which one is favored by the population? This chapter is to explore and analyze the strategy selection problem in several typical evolutionary dynamic models of networked games. In detail, firstly the concept of networked games is introduced together with several typical evolutionary dynamics models, including the birth-death process, the death-birth process, and the imitation dynamics. Then, several results of strategy selection conditions are reported for evolutionary dynamics of both two-player multi-strategy games and multi-player two-strategy games on networks. Moreover, these results are applied to the prisoner's dilemma game, the volunteer's dilemma game, and the public goods game to investigate the cooperation conditions in networked populations. The main aim of this chapter is to characterize the effect of interacting networks on strategy selection and more specifically on the evolution of cooperation.

17.1 Introduction

Game theory is an underlying paradigm of a number of fields of science [1, 2]. The abstract game models provide a unified framework to characterize the interplay among different choices in populations, for example, the interactive decision

S. Tan

College of Electrical and Information Engineering, Hunan University,
Changsha 410082, People's Republic of China
e-mail: shaolintan@hnu.edu.cn

J. Lü (✉)

Institute of Systems Science, Academy of Mathematics and Systems Science,
Chinese Academy of Sciences, Beijing 100190, People's Republic of China
e-mail: jhlu@iss.ac.cn

situations in economics, the interrelationship among different species in biology, and the mutual activity of behaviors in social sciences [3–6].

Equilibrium selection is one of the most fundamental issues in game theory [7]. It aims at addressing the way how individuals achieve a stable strategy equilibrium. Early mathematical theory of equilibrium selection is mainly based on the assumption of perfect rationality of players. That is, each individual will choose the strategy which optimizes the utility of himself with the expectation of strategies of others. This reasoning methodology, though is very successful, has been in dispute since perfect rationality seems to be an unrealistic and ill-defined concept [8].

Evolutionary game theory addresses the equilibrium selection problem with a completely different approach. In evolutionary games, successful strategies propagate by reproduction or imitation in a population. And the strategy equilibrium is selected by a dynamic process of individuals [9, 10]. Compared with the classical game theory, evolutionary game theory abandons the assumption of perfection rationality. Instead, strategy updating rules are proposed to characterize the strategy selection of individuals.

Evolutionary game theory was used to describe the diffusion of successful phenotypes and behaviors in biological species initially [11, 12]. Lately, it has been widely applied to explore various evolutionary processes in social, economical, and technological systems [13–15]. Over the last decades, the pattern of interactions underlying complex systems has drawn an extensive attention. The theory of complex networks has been developed for an advanced understanding of various natural and social systems [16–19]. In particular, kinds of social networks have been obtained. And a variety of mathematical models, such as the scale-free and small-world networks, have been established to characterize the interaction pattern of individuals [20, 21]. To investigate the problem of strategy selection in these networked systems, networked evolutionary games are introduced and become an area of great interest.

In networked evolutionary games, individuals connect as a network: each node represents an individual and an edge represents an interrelationship between the two individuals at the endpoints of the edge. Individuals play game only with their neighbors to gain payoffs. Generally, two types of interaction modes are frequently investigated. The first type is pairwise interactions, where individuals interact in pair to acquire payoffs. In this case, two-player games are played between individuals and the payoff of an individual only depends on the strategies of his neighbors [22]. The other type is group interactions, where each individual together with all his neighbors forms a group and the payoff is determined by strategies of the whole group. For group interactions, multi-player games are played among each group and the payoff of an individual depends not only on the strategies of his neighbors but also on those of the neighbors of his neighbors [23, 24].

The strategy updating rule plays an essential part in equilibrium selection in networked evolutionary games. For evolutionary games in well-mixed populations, the evolutionary dynamics of population strategies is described in terms of strategy frequencies. Yet, such a macroscopic description is inadequate for networked evolutionary games. To characterize how individuals update their strategies based on the

information of surrounding neighborhood, various microscopic agent-based strategy updating rules have been proposed. These strategy updating rules either mimic the Darwinian evolutionary process, taking mutation and selection as the core force driving the evolution of population strategy, or consult to social experiences, using behavior rules such as imitation or comparison to adjust strategies [25–27].

Networked evolutionary games can exhibit behaviors which do not appear in traditional evolutionary games. In the evolutionary game dynamics on complex networks, interactions happen in a short range and the strategy updating process is distributed. Such a situation allows a variety of strategy associates. For example, it is shown that various spatial patterns of strategy distribution can emerge in evolutionary games on spatial grids [28]. Moreover, in traditional evolutionary games, cooperation can never favored over defection. Yet, in networked evolutionary games, cooperation may be promoted through network reciprocity [29–31].

This chapter focuses on the strategy selection problem in networked evolutionary games. The concerned question is: given several strategies, which strategy is favored by the population and how the population select their favorite strategy through microscopic individual strategy updating? Recent works have revealed that strategy selection depends not only on the utility function of strategies but also on the interaction structure of the population. Several strategy selection conditions have been derived for different types of games, networks and updating rules [32–36]. A tutorial review of these progresses will be given in this chapter.

Understanding the evolution of cooperation is a primary objective of evolutionary game theory. Cooperation is of essential importance for social and biological systems. Cooperation among individuals can greatly cut down on wastage and promote the common welfare of the entire population, yet it is costly and can be easily exploited by selfish behaviors [23, 37, 38]. To solve the puzzle of cooperation, various cooperation mechanisms have been proposed [39, 40]. Recently, the heterogeneous interactions among individuals have been regarded as positive factors in promotion of cooperation. In complex networks, cooperators could survive by forming cooperative clusters to resist the exploitation of selfish individuals [41–43]. In this chapter, the strategy selection conditions are further applied to investigate the evolution of cooperation in networked populations. The conditions for favor of cooperation have been derived for the prisoner’s dilemma game, the volunteer dilemma game, and the public goods game.

The setup of the present chapter is as follows. Section 17.2 gives a brief introduction of the networked game and several typical agent-based evolutionary dynamics, including the birth-death process, death-birth process and imitation dynamics. Section 17.3 presents the strategy selection conditions for evolutionary dynamics with different games and networks. To conclude, in Sect. 17.4, a case study is given to illustrate how strategy selection in populations is jointly influenced by interaction modes, the utilities of strategies, and the interaction network.

17.2 Evolutionary Dynamics of Networked Games

Compared with traditional games, networked evolutionary games possess two distinctive features: Firstly, the specific interacting structure among individuals is taken into account; and secondly, the requirement of perfect rationality on each individual is abandoned. In networked evolutionary games, individuals only interact with their neighbors to gain payoffs. Moreover, without the assumption of perfect rationality, various individual-based strategy updating dynamics are proposed to characterize the decision-making behaviors of individuals. In this section, we present necessary definitions and concepts of networked games together with several typical evolutionary dynamic models that describe the evolution of population strategies in networked games.

17.2.1 Networked Games

A game is an abstract formulation of an interactive decision situation with explicit stipulations of the players engaged in, their available strategies or actions, and the payoffs for each strategy profile. A normal definition of game is as follows.

Definition 17.1 A game is a triple $\Gamma = (V, \{S_v | v \in V\}, \{u_v | v \in V\})$, where

- (1) V is a set of players,
- (2) S_v is the set of possible strategies (or actions) of player v ,
- (3) and $u_v : \prod_{i \in V} S_i \rightarrow R$ is the payoff function of player v .

When $|V| = N$, Γ is called N -player game. And when $|S_v|$ is finite, it is called a finite game. If the strategy set S_v and payoff function u_v are identical for all player v (that is, for all v , $S_v = S$, $u_v = u$), and the payoff function u is irrelevant to the identities of the players, then Γ is referred to as a symmetric game. Throughout this chapter, the discussion only focuses on N -player finite symmetric games.

The above game formulation depicts three key issues of the decision-making situations. Yet, it is too general to give a more meticulous description of how individual interacts with each other. In realistic multi-player systems, players do not always interact with all the other players. Instead, they only interact with those players in their neighborhood. To capture such a decision situation, networked games are introduced with an additional stipulation of how a player's payoff depends on a subset instead of all of other players.

Let an undirected network $G = (V, E)$ denote the interaction structure of a population, where the players and interactions among players are represented by the node set $V = \{1, 2, \dots, N\}$ and edge set E , respectively. And let $N(i) = \{j | (i, j) \in E, j \in V\}$ denote the neighborhood of node i . A normal definition of networked game is as follows [44].

Definition 17.2 A networked game is a triple $\Gamma = (G = (V, E), \{S_v | v \in V\}, \{u_v | v \in V\})$, where

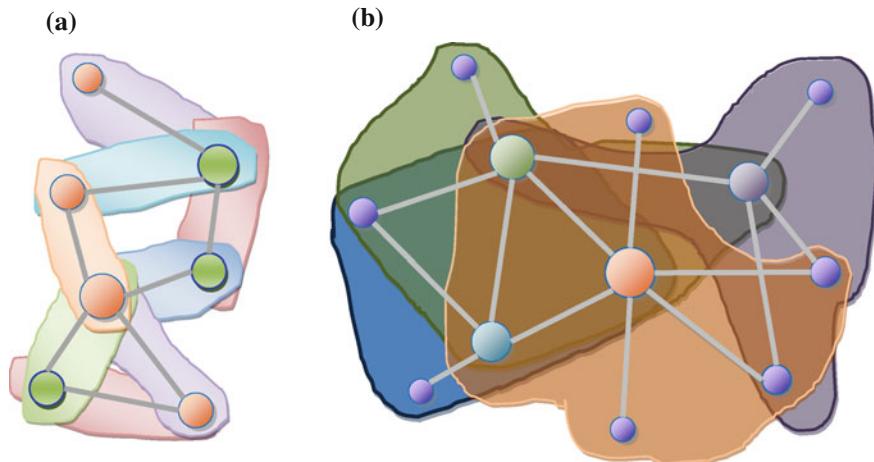


Fig. 17.1 Two typical game interactions on networks. (*left*) Pairwise interactions. Individuals in the network interact in pairwise games to gain payoffs. (*right*) Group interactions. Individuals together with their neighbors form groups and engage in a multi-player game

- (1) $G = (V, E)$ is an undirected network,
- (2) S_v is the set of possible strategies (or actions) of player v ,
- (3) and $u_v : \prod_{i \in N(v) \cup \{v\}} S_i \rightarrow R$ is the payoff function of player v .

Compared with general games, the difference of networked games lies in that the payoff function of each player only depends on the strategies of its neighbors now. Generally, the payoff function of each node can be distinct from one another in the networked game. Yet, for simplicity, two common interaction modes are mainly considered in networked evolutionary games (see Fig. 17.1). The first type is pairwise interactions. In this case, each pair of players engages in a two-player game, and the payoff of a node is the sum of payoffs acquired from the game interaction with each neighbor. The other type is group interaction, in which case, each node together with their neighbors forms a group and its payoff is determined by strategies of the involved groups.

17.2.2 Agent-Based Evolutionary Dynamics

The agent-based evolutionary dynamics describes how agents update their strategies with the information of payoffs and strategies of the surrounding environment. Such a strategy-updating process is usually very complex in realistic problems. Yet, based on different assumptions on the players, a large variety of strategy-updating rules have been developed from different backgrounds. In this chapter, we only concentrate on several typical evolutionary dynamics.

17.2.2.1 The Darwinian Updating Rules

The Darwinian updating rules mimic the evolutionary process of biological populations. Replication, selection, and mutation are utilized as three basic mechanisms to govern the evolution of population strategies in networked games. Through the evolutionary process, those strategies which bring individuals higher payoffs will be preserved and those disadvantageous strategies will be eliminated, which finally provides an answer to the question which strategy is favored to the population.

In the microscopic dynamics of Darwinian updating, the payoffs obtained from the game interactions are interpreted as fitness. An exponential formula

$$\text{fitness} = \exp(w \times \text{payoff}) \quad (17.1)$$

is commonly used to determine each individual's fitness. Here, w is a parameter adjusting the intensity of selection, where $0 \leq w < \infty$. When the selection intensity is small, a linearized equation

$$\text{fitness} = 1 + w \times \text{payoff} \quad (17.2)$$

is usually used as an alteration of Eq. (17.1). For $w = 0$, all individuals possess the same fitness. In this case, the evolutionary dynamics of individuals's strategies is irrelevant to the game interactions and it is dominated by random drift [46]. For $w \rightarrow 0$, the payoff acquired from game interactions takes an extremely small part of the fitness. And thus, all individuals have almost equal fitness, which is called weak selection [45].

Figure 17.2 shows two typical Darwinian updating rules: the ‘birth-death’ (BD) process and the ‘death-birth’ (DB) process [47–49]. In the BD process, at each time step, one individual is firstly selected out of the whole population for reproduction with a probability proportional to its fitness. Then, a neighbor of the selected individual is randomly chosen. In the last, the selected individual reproduces an identical copy of itself and replaces the chosen neighbor with the copy. While in the DB process, the order of birth and death is interchanged. Firstly, one individual is chosen randomly and then eliminated out of the population. Then, a neighboring individual of the chosen one is selected out with a probability proportional to fitness. In the last, the selected neighbor reproduces a identical copy of itself and replaces the eliminated individual.

During the reproduction process, mutation may happen with some probability. In this case, the offspring may not always possess a strategy which is identical to its parent. Let μ denote the mutation rate, where $0 \leq \mu < 1$. Then in the reproduction process, the offspring adopts a random strategy with probability μ . Mutation enlarges the strategy space and is regarded as an important force that draws the population out of those local fitness valleys in the evolutionary games.

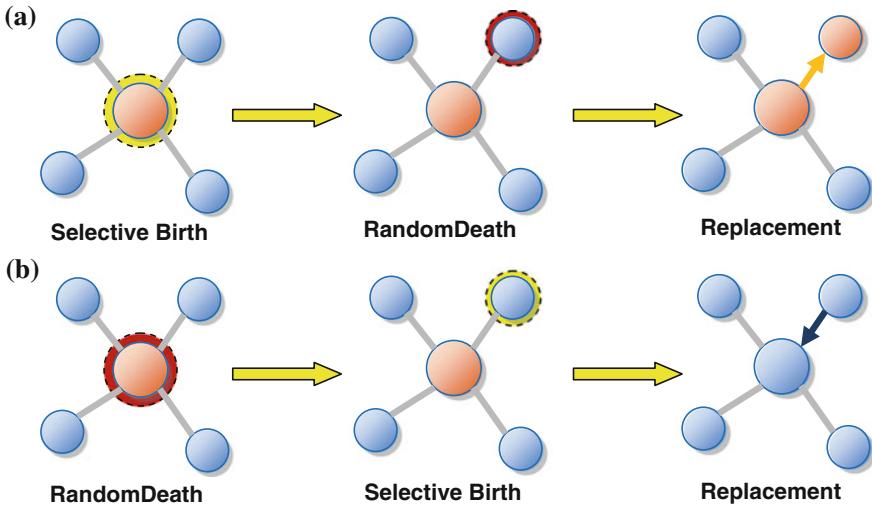


Fig. 17.2 Illustration of two typical Darwinian updating rules on networks. (*upper*) Birth-death process; (*bottom*) Death-birth process

17.2.2.2 The Socialized Updating Rules

In the above Darwinian dynamics, strategy updates are driven by exterior mechanisms, such as selection and mutation. Such an objective description is adequate, if individuals do not have subjective initiative and practical abilities to update their strategies. However, when we consider strategy updating of human beings, who can alter their strategies with a desire of better payoffs, socialized dynamics are more proper.

There exist various ways how individuals update their strategies on the basis of local environment. The strategy change may be determined by memory of past information of payoffs and strategies, and it can be also based on anticipation of possible future changes of its neighbors' strategies. Here, we mainly focus on a common memoryless myopic system: the imitation dynamics, where the updating rule only depends on the current state of population [50].

Imitation is a commonly utilized explanation of many real-world collective behaviors. When information about the underlying game is limited, the players often update strategies through imitating the strategies of those successful neighbors. Let $x_i(t)$, $f_i(t)$ denote the strategy and payoff of player i at step t , respectively. Suppose the strategy space is continuous. A representative mathematical model of the imitation process is

$$\begin{aligned} x_i(t+1) &= x_i(t) + \Delta_i(t), \quad i = 1, 2, \dots, N, \\ \Delta_i(t) &= \sum_{k=1}^N \frac{a_{ik}}{2d_i} [\tanh(w(f_k - f_i)) + 1][x_k(t) - x_i(t)]. \end{aligned} \quad (17.3)$$

In the above equation, the parameter w depicts the selection intensity. For $w = 0$, the above equation becomes

$$x_i(t+1) = \frac{1}{2}x_i(t) + \sum_{k=1}^N \frac{a_{ik}}{2d_i} x_k(t). \quad (17.4)$$

It indicates that each player imitates all its neighbors with the same probability. Yet, if the selection intensity w is large enough, players only imitate those successful neighbors, i.e. neighbors with higher payoffs, and the probability that they imitate those neighbors with lower payoffs tends to zero.

17.3 Strategy Selection in Networked Evolutionary Games

“Strategy selection” is introduced to provide a solution concept to evolutionary dynamics of networked games. Similar to the concept Nash equilibrium, it aims at giving an answer to the question which strategy is favored by the population. Yet, unlike the static strategy equilibrium in classical games, the successful strategy in evolutionary games is selected and discriminated through dynamic strategy-updating process of all players.

Generally, let the game dynamics start evolving from an unbiased state, that is, the initial strategy distribution is symmetric for all strategies. Then, record the stationary distribution of strategies in the evolutionary dynamics. If the frequency of a strategy is larger than its initial value, then it is called that the strategy is favored by selection. And if the frequency of one strategy is larger than that of another in the stationary distribution, then it is said that the former strategy is favored over the latter one. Obviously, strategy selection depends not only on the game but also on the updating rules and interaction structure of the population [34].

17.3.1 Strategy Selection in Darwinian Dynamics

17.3.1.1 Two-Player Two-Strategy Game

Two-player two-strategy game is the simplest type of game. Let $V = \{1, 2\}$ denote the two players and $S = \{A, B\}$ the two strategies, a two-player two-strategy symmetric game can be represented by

	A	B
A	a	b
B	c	d

(17.5)

Here, a, b, c and d are player 1's payoff for the strategy profile (A, A) , (A, B) , (B, A) , and (B, B) , respectively.

As depicted in Sect. 17.2, individuals play the above game with each neighbor and the acquired payoff is summed up and interpreted as their fitness. Then, based on the fitness landscape, the population updates their strategies with the Darwinian updating rule. Repeat of the above operations leads to an evolutionary dynamics of the networked games.

In an appropriate manner, the above evolutionary process defines a finite-state discrete-time Markov chain. Through repeated updating, the network of strategies is transformed from one state to another at each step. If the mutation rate $\mu = 0$, then the corresponding Markov chain will be finally absorbed at an all- A or all- B state, as shown in Fig. 17.3. Let ρ_A be the fixation probability that a single A -player generates a lineage of offsprings that take over the whole population after invading a network of B -player from a random location. Vice versa, let ρ_B be the fixation probability that a single B -player generates a lineage of offsprings that take over the whole population after invading a network of A -player from a random location. When the mutation rate $\mu = 0$, the strategy selection problem reduces to comparison of the fixation probability ρ_A and ρ_B . The condition that strategy A is favored over strategy B becomes

$$\rho_A > \rho_B. \quad (17.6)$$

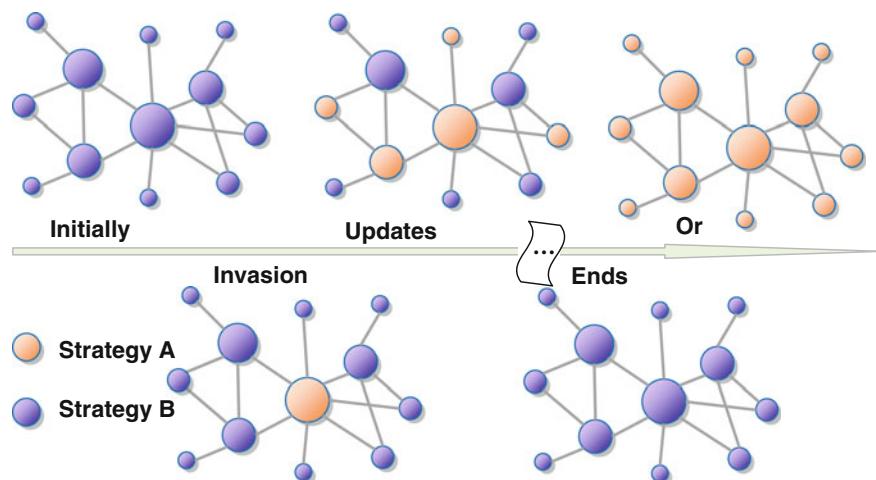


Fig. 17.3 Illustration of the evolutionary process of strategies on complex networks. Initially, the network is composed of B -players. The evolutionary process begins with a random invasion of an A -player. Based on the fitness landscape and updating rule, the network transforms from one state to another. If mutation is not considered, then the population will reach fixation at all- A or all- B state [36]

If the mutation rate $\mu > 0$, then the corresponding Markov chain of the evolutionary process has a stationary distribution. In this case, strategy A is said to be favored over strategy B if the expectation of the frequency of A -players in the stationary distribution is larger than 0.5 in the population [33].

In well-mixed populations (i.e. complete networks), each individual interacts with all the other $N - 1$ individuals. In this case, the strategy selection condition can be easily derived. Indeed, it was proved in [51, 52] that strategy A is favored over strategy B if

$$\frac{N-2}{N}a + b > c + \frac{N-2}{N}. \quad (17.7)$$

The above condition holds for both BD and DB processes with any selection intensity and mutation rate. In the limit $N \rightarrow \infty$, the above condition becomes $a + b > c + d$, which is the standard condition of risk-dominance.

However, in general networks, it becomes much harder to determine which strategy is favored in the evolutionary dynamics of networked games. The main difficulty lies in that the distribution of strategies on networks allows a large number of configurations, which makes the size of the state space of the corresponding Markov chain increase exponentially with the network size. Nevertheless, in [33], Tarnita et al. proved that in the limit of weak selection (i.e. the selection intensity $w \rightarrow 0$), strategy A is favored over strategy B if and only if

$$\sigma a + b > c + \sigma d. \quad (17.8)$$

Here, σ is called the structural coefficient which depends on the network topology, updating rule and mutation rate but is irrelevant to the utilities a, b, c , and d .

Table 17.1 lists the values of σ for both the BD and DB processes on several simple networks. The results hold for any mutation rate in star graphs under the DB process and complete networks under both BD and DB processes. In the other cases, the results are valid only for low mutation (i.e. the mutation rate $\mu \rightarrow 0$).

The above explicit conditions of strategy selection can be derived for highly symmetric networks. For general heterogeneous networks, yet, only approximate results are obtained. Consider a network with a node degree sequence (d_1, d_2, \dots, d_N) .

Table 17.1 The structural coefficients σ of several simple networks governed by pairwise interactions

Graph	σ for BD process	σ for DB process
Complete graph	$(N-2)/N$	$(N-2)/N$
Cycle graph	$(N-2)/N$	$(3N-8)/N$
Star graph	$(N^3 - 4N^2 + 8N - 8)/(N^3 - 2N^2 + 8)$	1
Regular graph of degree k	$(N-2)/N$	$[(k+1)N - 4k]/[(k-1)N]$

Here, N denotes the network order

Denote $k = \sum_{i=1}^N d_i / N$ the average degree of the network. Ohtsuki et al. found that

$$\sigma = \frac{k+1}{k-1} \quad (17.9)$$

makes an excellent approximation for the DB process on general networks [53]. Denote $k_{nn,i} = \frac{1}{d_i} \sum_{v \in N(i)} d_v$ the average nearest-neighbors degree of node i and $k_{nn} = \sum_{i=1}^N k_{nn,i} / N$ the average nearest-neighbors degree of the network. Konno found that

$$\sigma = \frac{k_{nn} + 1}{k_{nn} - 1} \quad (17.10)$$

makes a better approximation for the DB process on general uncorrelated networks [54].

The above results can be directly applied to explore the evolution of cooperation in structured populations. Consider a simplified prisoner's dilemma game with a payoff matrix

	Cooperation	Defection
Cooperation	$b - c$	$-c$
Defection	b	0

(17.11)

In this game, a cooperator contributes a benefit b to its partner at a cost of c of himself, while a defector pays no cost and contributes no benefit, where $b > c > 0$. For perfect rational players, the defection strategy is the best personal choice, which yet leads to the worst outcome. However, according to Eq. (17.8), the cooperation strategy can be favored over the defection strategy in evolutionary game dynamics if and only if

$$\frac{b}{c} > \frac{\sigma + 1}{\sigma - 1}, \quad (17.12)$$

or equivalently

$$\sigma > \frac{b+c}{b-c}. \quad (17.13)$$

Thus, under the BD process, cooperation is never favored. Yet, under the DB process, cooperation is favored over defection by large cycle networks if the benefit-to-cost rate b/c is larger than 2. And for general networks, the condition of favoring cooperation is $b/c > k_{nn}$.

17.3.1.2 N -player Two-Strategy Game

A general N -player two-strategy symmetric game can be represented as follows:

Number of opposing A players	0	1	\dots	k	\dots	$N - 1$
A	a_0	a_1	\dots	a_k	\dots	a_{N-1}
B	b_0	b_1	\dots	b_k	\dots	b_{N-1}

(17.14)

That is, in a group of N players, if there are k A -players, then an A -player acquires a payoff a_{k-1} , while a B -player gains a payoff b_k .

Consider the evolutionary dynamics of the above game on a network. In detail, each node together with all its neighbors forms a group and in each group the above N -player two-strategy symmetric game is played. Hence, a player with k neighbors takes part in a total of $k + 1$ groups. And the payoff of each individual is the sum of obtained payoffs from all the groups it participates in. The question is which strategy is favored by the population under the BD and DB process.

In [36], the strategy selection conditions are derived for three simple networks in the case of low mutation and weak selection.

- (i) Complete graphs. For both the BD and DB processes, the condition that strategy A is favored over B is

$$\sum_{i=1}^{N-1} a_{i-1} > \sum_{i=1}^{N-1} b_i. \quad (17.15)$$

Here, N is the number of nodes of the complete graph.

- (ii) Cycle graphs. On cycle graphs, each group contains only three individuals. For the BD process, strategy A is favored over strategy B if and only if

$$N(a_0 + a_1 + a_2) - 3a_2 > N(b_0 + b_1 + b_2) - 3b_0. \quad (17.16)$$

And for the DB process, strategy A is favored over B if and only if

$$(N+1)a_0 + (2N-2)a_1 + (3N-11)a_2 > (3N-11)b_0 + (2N-2)b_1 + (N+1)b_2. \quad (17.17)$$

Here, N is the number of nodes of the cycle.

- (iii) Star graphs. On a star graph with N nodes, there are two types of interacting groups. The first type of group focuses on the leaf, containing only two individuals: the leaf and the center. And the second type of group focuses on the center, containing all the N individuals. Suppose that the first group plays game (17.5) and the second group plays game (17.14). In the limit $N \rightarrow \infty$, the condition for strategy A to be favored over B in the BD process is

$$a + b > c + d, \quad (17.18)$$

and in the DB process, the condition becomes:

$$a + b + \frac{a_0 + a_{N-2}}{2} + \frac{1}{N} \sum_{i=1}^{N-3} a_i > c + d + \frac{b_1 + b_{N-1}}{2} + \frac{1}{N} \sum_{i=1}^{N-3} b_{i+1}. \quad (17.19)$$

The above results can be directly applied to explore the evolution of volunteer cooperation in structured populations. Consider a volunteer's dilemma game with the following form:

Number of opposing cooperation players	0	1	\dots	k	\dots	$N-1$
Cooperation	$-c$	$-c$	\dots	$-c$	\dots	$-c$
Defection	$-a$	0	\dots	0	\dots	0

(17.20)

In this game, individuals either take a volunteer action (i.e. cooperation) to contribute a public good or not to (i.e. defection). The cost of volunteering is c , and if nobody takes the voluntary action, then everybody will lose a . Here, $a > c > 0$.

According to the obtained strategy selection conditions, the conditions for favor of cooperation in the volunteer's dilemma game can be derived for complete networks, cycles and stars, as shown in Table 17.2. It can be found that cooperation is never favored on complete networks, yet it is favored over defection on cycle and star graphs given that the cost of cooperation is less than one-half or one-third cost of entire defection.

Now consider a specified type of N -player two-strategy game, in which the payoffs of the A -player and B -player in a group of i A -players and j B -players are

$$\begin{aligned} P_A &= (ia + jb)/(i + j), \\ P_B &= (ic + jd)/(i + j), \end{aligned} \quad (17.21)$$

respectively. In this case, the strategy selection condition is the same as Eq. (17.8). Yet, the parameter σ is different, as shown in Table 17.3.

Let $a = (r - 1)$, $b = -1$, $c = r$, and $d = 0$ in Eq. (17.21), then we get a public goods game. In this game, individuals either contribute a unit amount to the public pool (cooperation) or not to (defection). The total contributions are then multiplied by a factor $r > 1$ and then equally distributed among all the individuals. Thus, in a group

Table 17.2 The conditions for evolution of cooperation in the volunteer's dilemma game

Graph	BD process	DB process
Complete graph	$c < 0$	$c < 0$
Cycle graph	$3c < a$	$2c < a$
Star graph	$2c < a$	$3c < a$

Here, the network size N is large enough, that is, $N \rightarrow \infty$ [36]

Table 17.3 The structural coefficients σ of three simple networks governed by group interactions

Graph	$\sigma(\text{BD})$	$\sigma(\text{DB})$
Complete graph	1	1
Cycle graph	$(2N - 3)/N$	$(7N - 18)/(2N)$
Star graph	$(3N^3 - 6N^2 + 4N + 8)/(N^3 + 2N^2 - 12N + 24)$	5/3

Here, N denotes the network size [36]

of i cooperators and j defectors, a cooperator gains a payoff $[(r - 1)i - j]/(i + j)$, and a defector gets a payoff $ri/(i + j)$.

According to the strategy selection condition, in the public goods game, cooperation is favored over defection if and only if

$$\sigma > \frac{r + 1}{r - 1} \quad (17.22)$$

or equivalently

$$r > \frac{\sigma + 1}{\sigma - 1}. \quad (17.23)$$

Thus, from Table 17.3, cooperation is never favored in complete networks, yet it can emerge in cycle and star networks for both BD and DB processes.

17.3.1.3 Two-Player M -strategy Game

Let $P = \{1, 2\}$ denote the two players and $S = \{s_1, s_2, \dots, s_M\}$ the set of strategies, a two-player M -strategy symmetric game can be fully characterized by a payoff matrix $U = (u_{ij})_{M \times M}$. Here, the matrix element u_{ij} denotes player 1's payoff for a strategy profile (s_i, s_j) .

Consider the Darwinian evolutionary process of the above game. For positive mutation rate $\mu > 0$ and weak selection, all strategies will have almost equal frequency, $1/M$, in the stationary distribution. A strategy is said to be favored by selection, if its expected frequency exceeds $1/M$. Otherwise, it is said to be opposed by selection.

In [55], Antal et al. has given a complete characterization of the strategy selection in well-mixed populations. Denote $\bar{u}_{k*} = (1/M) \sum_{i=1}^M u_{ki}$ the average payoff of strategy k , $\bar{u}_{**} = (1/M) \sum_{i=1}^M u_{ii}$ the average payoff when both individuals adopts the same strategy, $\bar{u}_{*k} = (1/M) \sum_{i=1}^M u_{ik}$ the average payoff when the opponent adopts strategy k , and $\bar{u} = (1/M^2) \sum_{i=1}^M \sum_{j=1}^M u_{ij}$ the average payoff in the population. It is proved in [55] that the condition for strategy k to be favored by selection is fully determined by quantities of the above variances. In detail, strategy k is favored by selection if

$$L_k + N\mu H_k > 0, \quad (17.24)$$

where N is the population size, μ is the mutation rate, and

$$L_k = u_{kk} + \bar{u}_{k*} - \bar{u}_{*k} - \bar{u}_{**}, \quad (17.25)$$

$$H_k = \bar{u}_{k*} - \bar{u}. \quad (17.26)$$

Moreover, the condition for strategy k to be favored over strategy l is

$$L_k + NuH_k > L_l + N\mu H_l. \quad (17.27)$$

The above conditions only hold for large well-mixed populations. Regarding structured populations, in [56], Tarnita et al. proved that strategy k is favored by selection if

$$(\sigma_1 u_{kk} + \bar{u}_{k*} - \bar{u}_{*k} - \sigma_1 \bar{u}_{**}) + \sigma_2 (\bar{u}_{k*} - \bar{u}) > 0. \quad (17.28)$$

Here, the parameter σ_1 and σ_2 are called structured coefficients that depend on the population structure, the updating rule and mutation rate but are irrelevant with the number and utilities of strategies. Furthermore, if the mutation rate is low (i.e. $\mu \rightarrow 0$), it is shown that $\sigma_2 \rightarrow 0$. In this case, the condition that strategy k is favored by selection becomes

$$\sigma_0 u_{kk} + \bar{u}_{k*} > \bar{u}_{*k} + \sigma_0 \bar{u}_{**} \quad (17.29)$$

where $\sigma_0 = (2\sigma_1 + \sigma_2)/(2 + \sigma_2)$ is just the structural coefficient which has been calculated in evolutionary dynamics of two-strategy games in structured populations.

17.3.2 Strategy Selection in the Imitation Dynamics

Now, we turn to investigate the strategy selection problem in imitation dynamics of networked games. Without loss of generality, consider the two-strategy two-player game (17.5). Suppose that player i adopts strategy A with probability x_i and strategy B with probability $1 - x_i$, where $0 \leq x_i \leq 1$. Then, the payoff of each player on the network is given by

$$f_i = \sum_{k \in N(i)} (x_i, 1 - x_i) \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x_k \\ 1 - x_k \end{pmatrix}, \quad (17.30)$$

where $N(i)$ denotes the neighbors of player i .

Starting from given initial states $x_i(0) \in [0, 1]$, if the underlying interaction network is connected, then under the imitation dynamics (17.3), the population state will reach consensus. That is, $\lim_{t \rightarrow \infty} x_i(t) = x^*$ for all $i = 1, 2, \dots, N$. Moreover, if the selection intensity $w = 0$, then the population state converges to

$$\bar{x} = \left(\sum_{j=1}^N d_j \right)^{-1} \sum_{i=1}^N d_i x_i(0), \quad (17.31)$$

It is said that selection favors strategy A over strategy B if the consensus state x^* is larger than \bar{x} .

In [50], it is proved that cooperation is never favored over defection under the imitation dynamics of the prisoner's dilemma game (17.11) in well-mixed populations. Yet, for structured populations, such as cycles, stars, and some other types of networks, it is shown that cooperation may be favored given proper initial conditions. Figure 17.4 presents the evolution of strategies in different networks. It can be found that the average cooperation probability increases by some extent with selection as compared with that without selection. Though it has been commonly known that the population structure can greatly affect the evolution of cooperation in the imitation dynamics, it still lacks complete comprehension of which structural characteristics mainly determine the effect.

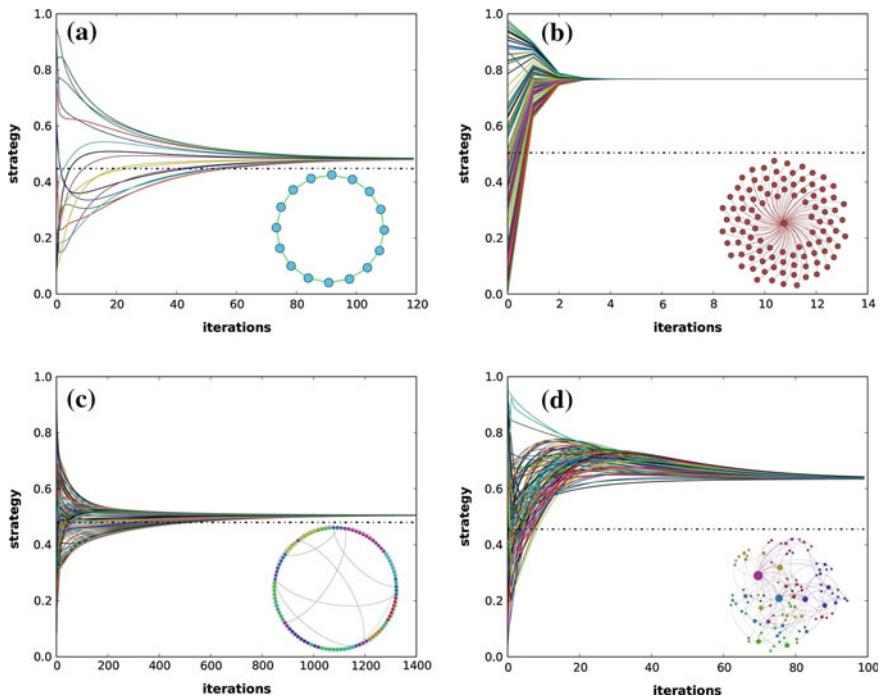


Fig. 17.4 Evolution of cooperation under the imitation dynamics on **a** a cycle with 16 nodes; **b** a star with 100 nodes; **c** a Newman-Watts-Strogatz small-world network with 100 nodes; and **d** a Barabási-Albert scale-free network with 100 nodes. The *dot dashed line* is the initial cooperation level of the population [50]

17.4 Case Study

In this section, a simple example is used to show how strategy selection is determined by the joint actions of the interaction mode, the utilities of strategies, the updating rule, and the interaction network.

Consider a game with a payoff matrix as follows:

$$\begin{array}{c|cc} & C & D \\ \hline C & 1 & 0 \\ D & c & 0 \end{array}. \quad (17.32)$$

For perfect rational players, strategy C and strategy D will be chosen if $c \leq 1$ and $c > 1$, respectively. Now we wonder which strategy is favored by the population in evolutionary game dynamics.

Figure 17.5 shows the strategy selection conditions for the networked evolutionary games with different settings. Here, ‘PI’ and ‘GI’ are abbreviations of pairwise interactions and group interactions respectively. Under the group interactions, the payoff functions are shown in Eq. (17.21), where $a = 1$, $b = d = 0$, and $c > 0$ is a free parameter. The mutation rate is $\mu = 0$, thus, the question of strategy selection reduces to comparison of the fixation probabilities. Strategy C is favored over strategy D if $\rho_C > \rho_D$ and vice versa.

From the simulation results, it can be found that the conditions for favor of strategy C in networked evolutionary games generally differ from that in rational games. In evolutionary game dynamics on networks, strategy C is favored over D if and only

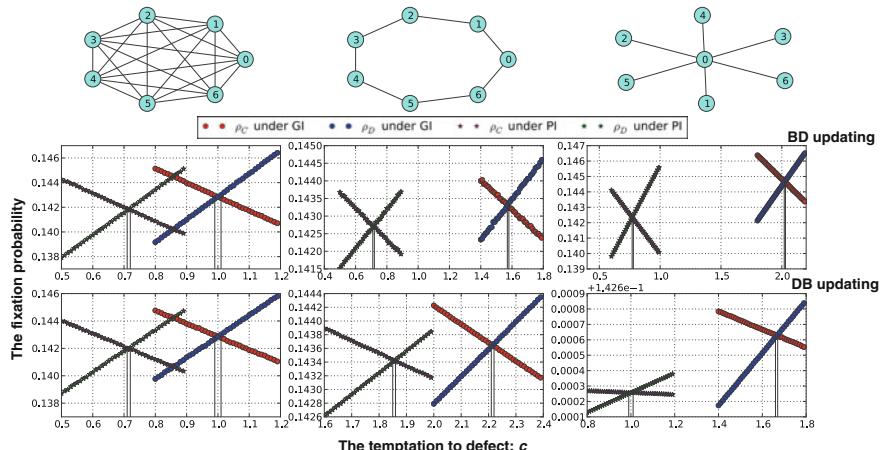


Fig. 17.5 Illustration of strategy selection in networked evolutionary games with different settings. The first row shows the type of network. The second and third rows display the fixation probabilities ρ_C and defection ρ_D . The selection strength is set as $w = 0.01$. The corresponding value of c at the intersection point gives an approximation of the structural coefficient [36]

if $c > \sigma$, where σ is a shift brought by the evolutionary dynamics and network structure. In some cases, the threshold value c_t for favor of cooperation shifts to values less than 1. While in some other cases, it shifts the opposite side. In particular, in the BD process on the star network with group interactions, the population favors strategy C even for $c = 2$, double of that in rational games.

17.5 Conclusions

Understanding the mechanism of strategy selection is one of the main goals in evolutionary game dynamics in structured populations. In this chapter, we have explored the strategy selection problem for evolutionary game dynamics with birth-death, death-birth, and imitation updating rules. The strategy selection conditions have been reported for both two-player multi-strategy games and multi-player two-strategy games on several simple networks. For complex networks, approximate and simulation results are presented to illustrate the effect of interaction structure on strategy selection.

The strategy selection conditions are applied to explore the evolution of cooperation in structured populations. The conditions for favor of cooperation have been reviewed for the prisoner's dilemma game, the volunteer's dilemma game, and the public goods game. The results could help us further understand the evolution of cooperative behaviors in different situations and backgrounds.

Acknowledgments This work was supported the 973 Project under Grant 2014CB845302, the Fundamental Research Funds for the Central Universities, the National Science and Technology Major Project of China under Grant 2014ZX10004-001-014, and the National Natural Science Foundation of China under Grant 61025017.

References

1. Smith, J.M.: *Evolution and The Theory of Games*. Cambridge University Press, Cambridge (1982)
2. Nowak, M.A.: *Evolutionary Dynamics: Exploring the Equation of Life*. Harvard University Press, Cambridge (2006)
3. Camerer, C.: *Behavioral Game Theory*. Princeton University Press, Princeton (2003)
4. Smith, J.M., Price, G.R.: The logic of animal conflict. *Nature* **246**, 15–18 (1973)
5. Axelrod, R., Hamilton, W.D.: The evolution of cooperation. *Science* **211**, 1390–1396 (1981)
6. Doebeli, M., Knowlton, N.: The evolution of interspecific mutualisms. *Proc. Natl. Acad. Sci. USA* **95**, 8676–8680 (1998)
7. Samuelson, L.: *Evolutionary Games and Equilibrium Selection*. MIT Press, Cambridge (1997)
8. Szabó, G., Fath, G.: Evolutionary games on graphs. *Phys. Rep.* **446**, 97–216 (2007)
9. Weibull, J.W.: *Evolutionary game theory*. The MIT Press, Cambridge (1995)
10. Ohtsuki, H.: Stochastic evolutionary dynamics of bimatrix games. *J. Theor. Biol.* **264**, 136–142 (2010)
11. Hofbauer, J., Sigmund, K.: *Evolutionary games and population dynamics*. Cambridge University Press, Cambridge (1998)

12. Nowak, M.A., Sigmund, K.: Evolutionary dynamics of biological games. *Science* **303**, 793–799 (2004)
13. Bentley, R.A., Hahn, M.W., Shennan, S.J.: Random drift and culture change. *Proc. R. Soc. B* **271**, 1443–1450 (2004)
14. Herzog, H.A., Bentley, R.A., Hahn, M.W.: Random drift and large shifts in popularity of dog breeds. *Proc. R. Soc. B* **271**, 353–356 (2004)
15. Ziman, J. (ed.): *Technological Innovation as An Evolutionary Process*. Cambridge University Press, Cambridge (2000)
16. Newman, M.E.J.: The structure and function of complex networks. *SIAM Rev.* **45**, 167–256 (2003)
17. Lu, J., Leung, H., Chen, G.: Complex dynamical networks: modeling, synchronization and control. *Dyn. continuous. Discret. Ser. B* **11a**, 70–77 (2004)
18. Lu, J., Chen, G.: A time-varying complex dynamical network model and its controlled synchronization criteria. *IEEE Trans. Autom. Control* **50**, 841–846 (2005)
19. Boccaletti, S., Latora, V., Moreno, Y., Chavez, M., Hwang, D.U.: Complex networks: structure and dynamics. *Phys. Rep.* **424**, 175–308 (2006)
20. Barabasi, A.L., Albert, R.: Emergence of scaling in random networks. *Science* **286**, 509–512 (1999)
21. Newman, M.E.J., Watts, D.J.: Renormalization group analysis of the small-world network model. *Phys. Lett. A* **263**, 341–346 (1999)
22. Nowak, M.A., Tarnita, C.E., Antal, T.: Evolutionary dynamics in structured populations. *Philos. Trans. R. Soc. B* **365**, 19–30 (2010)
23. Velicer, G.J., Yu, Y.N.: Evolution of novel cooperative swarming in the bacterium *Myxococcus xanthus*. *Nature* **425**, 75–78 (2003)
24. Perc, M., Gomez-Gardenes, J., Szolnoki, A., Floria, L.M., Moreno, Y.: Evolutionary dynamics of group interactions on structured populations: a review. *J. R. Soc. Interface* **10**, 20120997 (2013)
25. Ewens, W.J.(ed.): *Mathematical Population Genetics*. 2nd edn. Springer, New York (2004)
26. Tan, S., Lu, J., Chen, G., Hill, D.: When structure meets functions in evolutionary dynamics on complex networks. *IEEE Circuits Syst. Mag.* **14**, 36–50 (2014)
27. Wu, B., Zhou, D., Fu, F., Luo, Q., Wang, L.: Evolution of cooperation on stochastic dynamical networks. *PLoS ONE* **5**, e11187 (2010)
28. Nowak, M.A., May, R.M.: Evolutionary games and spatial chaos. *Nature* **359**, 826–829 (1992)
29. Johnson, D.D.P., Stopka, P., Knights, S.: Sociology—the puzzle of human cooperation. *Nature* **421**, 911–912 (2003)
30. Langer, P., Nowak, M.A., Hauert, C.: Spatial invasion of cooperation. *J. Theor. Biol.* **250**, 634–641 (2008)
31. Tan, S., Lu, J., Yu, X., Hill, D.: Evolution and maintenance of cooperation via inheritance of neighborhood relationship. *Chin. Sci. Bull.* **58**, 3491–3498 (2013)
32. Ohtsuki, H., Nowak, M.A.: Evolutionary games on cycles. *Proc. R. Soc. B* **273**, 2249–2256 (2006)
33. Tarnita, C.E., Ohtsuki, H., Antal, T., Fu, F., Nowak, M.A.: Strategy selection in structured populations. *J. Theor. Biol.* **259**, 570–581 (2009)
34. Tarnita, C.E., Antal, T., Ohtsuki, H., Nowak, M.A.: Evolutionary dynamics in set structured populations. *Proc. Natl. Acad. Sci.* **106**, 8601–8604 (2009)
35. Tang, C., Li, X., Cao, L., Zhan, J.: The σ lar of evolutionary dynamics in community-structured population. *J. Theor. Biol.* **306**, 1–6 (2012)
36. Tan, S., Feng, S., Wang, P., Chen, Y.: Strategy selection in evolutionary game dynamics on group interaction networks. *Bull. Math. Biol.* **76**, 2785–2805 (2014)
37. Rainey, P.B., Rainey, K.: Evolution of cooperation and conflict in experimental bacterial populations. *Nature* **425**, 72–74 (2003)
38. Griffin, A.S., West, S.A., Buckling, A.: Cooperation and competition in pathogenic bacteria. *Nature* **430**, 1024–1027 (2004)
39. Nowak, M.A., Sigmund, K.: Evolution of indirect reciprocity. *Nature* **427**, 1291–1298 (2005)

40. Nowak, M.A.: Five rules for the evolution of cooperation. *Science* **314**, 1560–1563 (2006)
41. Santos, F.C., Pacheco, J.M., Lenaerts, T.: Evolutionary dynamics of social dilemmas in structured heterogeneous populations. *Proc. Natl. Acad. Sci. USA* **103**, 3490–3494 (2006)
42. Cremer, J., Melbinger, A., Frey, E.: Growth dynamics and the evolution of cooperation in microbial populations. *Sci. Rep.* **2**, 281 (2012)
43. Santos, F.C., Santos, M.D., Pacheco, J.M.: Social diversity promotes the emergence of cooperation in public goods games. *Nature* **454**, 213–216 (2008)
44. Brandt, F., Fischer, F., Holzer, M.: Equilibria of graphical games with symmetries. *Theor. Comput. Sci.* **31**, 675–685 (2011)
45. Gokhale, C.S., Traulsen, A.: Evolutionary games in the multiverse. *Proc. Natl. Acad. Sci. USA* **107**, 5500–5504 (2010)
46. Tan, S., Lu, J., Hill, D.: Towards a theoretical framework for controlling random drift on complex networks. *IEEE Trans. Autom. Control* **60**, 576–582 (2015)
47. Tan, S., Lu, J.: Characterizing the effect of population heterogeneity on evolutionary dynamics on complex networks. *Sci. Rep.* **4**, 5034 (2014)
48. Broom, M., Rychtář, J.: An analysis of the fixation probability of a mutant on special class of non-directed graphs. *Proc. R. Soc. A* **464**, 2609–2627 (2009)
49. Lieberman, E., Hauert, C., Nowak, M.A.: Evolutionary dynamics on graphs. *Nature* **433**, 312–316 (2005)
50. Tan, S., Lu, J., Hu, Y., Ogorzalek, M.J.: Explore strategy selection in populations via a continuous networked game dynamics. In: Proceeding IEEE International Symposium Circuits System, Melbourne, Australia, June 1–5 (2014)
51. Nowak, M.A., Sasaki, A., Taylor, C., Fudenberg, D.: Emergence of cooperation and evolutionary stability in finite populations. *Nature* **428**, 646–650 (2004)
52. Taylor, C., Fudenberg, D., Sasaki, A., Nowak, M.A.: Evolutionary game dynamics in finite populations. *Bull. Math. Biol.* **66**, 1621–1644 (2004)
53. Ohtsuki, H., Hauert, C., Lieberman, E., Nowak, M.A.: A simple rule for the evolution of cooperation on graphs and social networks. *Nature* **441**, 502–505 (2006)
54. Konno, T.: A condition for cooperation in a game on complex networks. *J. Theor. Biol.* **269**, 224–233 (2011)
55. Antal, T., Traulsen, A., Ohtsuki, H., Tarnita, C.E., Nowak, M.A.: Mutation-selection equilibrium in games with multiple strategies. *J. Theor. Biol.* **258**, 614–622 (2009)
56. Tarnita, C.E., Wage, N., Nowak, M.A.: Multiple strategies in structured populations. *Proc. Natl. Acad. Sci. USA* **108**, 2334–2337 (2011)

Chapter 18

Network Analysis, Integration and Methods in Computational Biology: A Brief Survey on Recent Advances

Shihua Zhang

Abstract With rapid accumulation of high-throughput data, network has become one of key paradigms in computational biology for analyzing biological systems. In the past fifteen years, many types of molecular networks have been extensively investigated, demonstrating great potentials to discover basic functions and reveal essential mechanisms. More recently, network has played many new roles in multiple networked data and data types. In this chapter, we aim to survey the recent developments on topics related to biological networks and network-based data integration, with the special emphasis on the computational aspect. The contents of this survey covers network-based cancer stratification and cancer driver discovery from mutation data, network-based discovery of disease modules, multiple similarity network fusion, network-regularized data integration, module detection in multi-layer networks, topological analysis of a disease-age gene network, comparative analysis of multiple transcriptional-factor networks, etc.

18.1 Introduction

Since the end of last century, network has become an important paradigm for understanding how genes, proteins, and small molecules interact to form cellular systems [1, 2, 27]. It has been well recognized that interactions of individual components or networks are ultimately responsible for an organisms' form, organization and functions. With rapid development of biotechnologies, a number of high-throughput technologies have been developed for studying interactions of molecules. Molecular networks have been extensively studied and explored from various aspects of living organisms [2]. Those studies help biologists not only to understand complicated phenomena but also to elucidate the essential principles or fundamental mechanisms of cellular systems [27].

S. Zhang (✉)

National Center for Mathematics and Interdisciplinary Sciences,
Academy of Mathematics and Systems Science, Chinese Academy of Sciences,
Beijing 100190, People's Republic of China
e-mail: zsh@amss.ac.cn

From both biological and theoretical viewpoint, different interaction types among biomolecules or other objects form various types of networks or graphs. Detailed mathematical definition and biological meaning of these networks have been surveyed [1, 2]. In particular, the rapid development of complex network theory such as small-world property, right-skewed degree distribution, network transitivity property, network motif and community structure also accelerate our understanding of topological structure of biological networks [19, 27]. Naturally, the well-understood graph-theoretical concepts can be used systematically to explore the topology, organization, function, and evolution of biological networks. Such studies will no doubt deepen our knowledge at a system-wide level and further enhance biological insights on living organisms [2, 27].

In recent years, with rapid development of techniques and emerging large-scale programs (e.g., TCGA [18, 23], ENCODE [5]), network has played special roles in the integrative and comparative analysis of large-scale networked data and others. In this chapter, we aim to review the recent developments on topics related to biological networks and network-based data integration. The contents of this survey cover a series of studies such as network-based cancer stratification and cancer driver discovery from mutation data, network-based discovery of disease modules, multiple similarity network fusion, network-regularized data integration, module detection in multi-layer networks, topological analysis of a disease-age gene network, comparative analysis of multiple transcriptional-factor networks. We mainly highlight those new advances of recent network-based studies and discuss open challenging problems in this field.

18.2 Network-Based Stratification of Tumor Mutations

With the development of large-scale cancer genomic program [18, 23], cancer subtype discovery has become an important and challenging issue. Diverse genomic data types have been used for addressing this problem. One of these sources is the somatic mutation profile which are obtained by applying high-throughput sequencing to compare the genome or exome of a patient's tumor to that of the germline to identify mutations that have become enriched in the tumor cell population [8]. It is believed that such mutations contain the causal drivers of tumor progression. Thus, similarities and differences in mutations across patients become an important research object which may provide invaluable information for tumor stratification. However, tumor stratification based on the whole mutation profile has been more different due to its special data characteristics. Particularly, somatic mutation profiles are extremely sparse, unbalanced and heterogeneous. It has been comprehensively reported that it is very common that clinically identical patients share very few mutations. These features enable the traditional clustering techniques can not be used to somatic mutations like other data types such as expression or methylation for tumor stratification discovery.

In a very recent work, Hofree et al. [8] have developed a novel method for overcoming the above problems by integrating somatic mutation profiles with knowledge of the molecular interaction network architecture of human system. The key consideration is that cancer is not a disease of individual mutations or genes, but of combinations or cooperation of genes acting in key interacting pathways or a subregion of the molecular system which corresponds to hallmark processes such as cell proliferation and apoptosis. Thus, two tumors may not have any mutations in common, but they may share the same pathway or subnetwork affected by these mutations. They proposed a network-based stratification (NBS) method by using network knowledge and somatic mutation profile to stratify a cohort tumors into meaningful subtypes. They have applied this method to stratify cancers from TCGA into subtypes which have been shown to be biologically informative and have a strong association to clinical outcomes.

NBS method The NBS was designed to combine a genome-scale somatic mutation matrix and a gene interaction network for producing a robust partition of patients into subtypes. The somatic mutations for all patients are represented as a matrix of binary (1, 0) states on genes, in which a '1' or '0' indicates a gene for which mutation has occurred in the corresponding patient or not. For each patient, NBS projects the mutation profile onto a human gene interaction network. Next network propagation was adopted to spread the influence of each mutation over its network neighborhood (Fig. 18.1a). Specifically, let's denote the patient-by-gene mutation matrix as F_0 , and degree-normalized gene network adjacency matrix as A . The network propagation applies a random-walk like process on the network by

$$F_{t+1} = \alpha F_t A + (1 - \alpha) F_0,$$

where α is a tuning parameter controlling the diffusion range through the network. The 'network-smoothed' matrix of patients is clustered into a predefined number of subtypes via a network-regularized non-negative matrix factorization technique (Fig. 18.1b). This is formulated into a problem by minimizing the following objective function:

$$\min_{W, H \geq 0} \|F - WH\|_F^2 + \text{trace}(W^T K W),$$

where W and H form a decomposition of the patient versus gene matrix F (obtained from network smoothing procedure) such that W is basis matrix, and H is the loading matrix. The $\text{trace}(W^T K W)$ function constrains the basis vectors in W to consider the network connections. The term K is an adjacency matrix of a nearest neighbors network derived from the graph Laplacian of an influence distance matrix derived from the original network. A consensus clustering approach was further applied to improve robustness of clustering.

Very recently, we have adopted NBS to integrate key genetic and epigenetic features of tumor samples from multiple cancer types to uncover novel pan-cancer heterogeneity ([15] and unpublished work). The identified pan-cancer stratification is predictive of clinical outcomes, and different cancer patients falling into

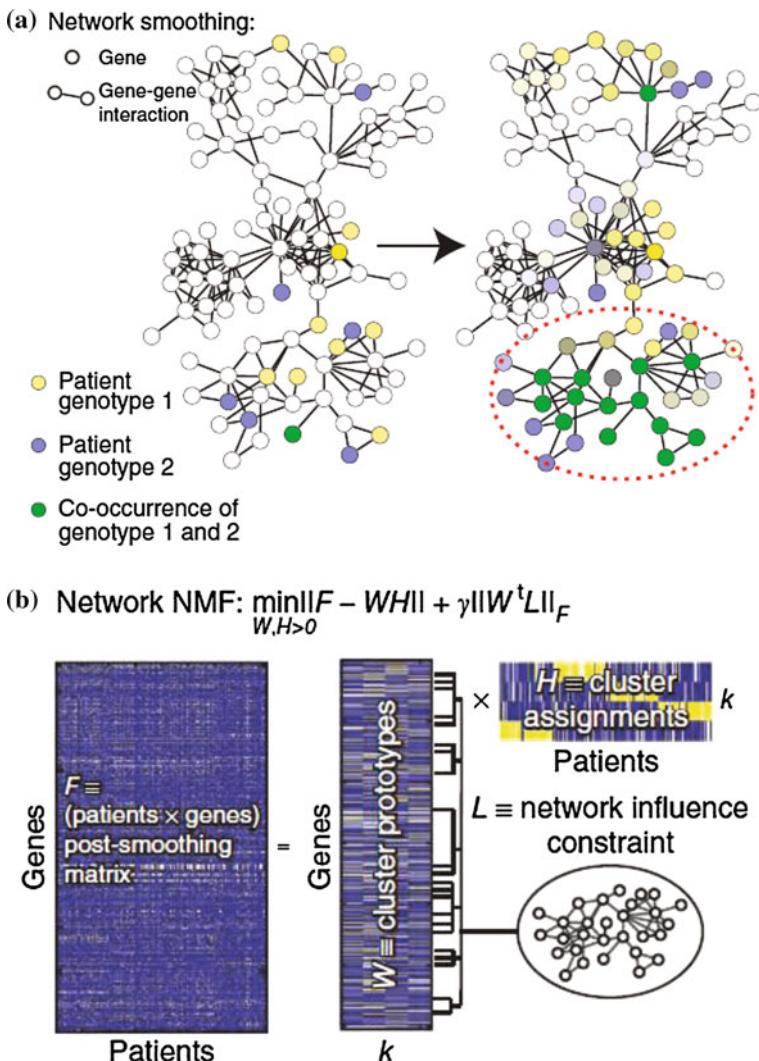


Fig. 18.1 Illustration of the two key steps of network-based stratification (NBS). **a** An toy example to demonstrate the network smoothing of patient somatic mutation profiles over a molecular interaction network. After smoothing, the mutational activity of a gene is a continuous value represented by intensity of yellow or blue for two patients; genes with high scores in both patients appear in green. **b** Clustering mutation profiles using a network-regularized non-negative matrix factorization (NMF) method. The input data matrix F is decomposed into the product of two matrices W and H . The decomposition attempts to minimize the objective function, k is a predefined number of subtypes. The figure is adapted from [8]

the same subgroup show consistent survival tendency or grade/stage severity. The identified subgroup-specific genomic alterations and altered networks demonstrate specific genomic characteristics and biological functional annotations. In summary, the network-based stratification paves a novel way to integrate mutation data for tumor classification.

18.3 Network-Based Discovery of Disease Modules

High-throughput experimental data such as protein-protein interaction, gene expression, DNA methylation and ChIP-seq data, are now widely explored to study the complicated behaviors of living organisms from various aspects at molecular level [5, 18, 23]. However, single type of data only provides limited information, e.g., protein-protein interaction data only tell possible interactions among proteins rather than when and where they interact with each other. Moreover, these data are very diverse from data type to scale. For example, protein-protein interaction data are generally quantified as discrete values, while gene expression data are usually expressed in the form of continuous real values. Therefore, how to integrate these heterogeneous data to elucidate the underlying biological mechanisms is an essential and challenging topic in computational biology and systems biology.

Generally, genes and their product proteins function in a concert rather than isolated manner. In particular, proteins interacting with other proteins, DNA, RNA and small molecules, form modules (e.g., complexes or pathways) to carry out cellular functions [27]. In contrast to individual components, it has been recognized that biomolecular networks or pathways are ultimately responsible to the forms and functions of living organisms, and can also reasonably explain the causes of various phenotypes. On the other hand, although protein interaction networks or pathways are available for many organisms based on accumulated protein interaction data and other experimental evidence, it is still a difficult task to identify active pathways or modules due to the changing conditions and environments in each living cell. In other words, while some genes (e.g., house-keeping genes) are constitutively expressed under various conditions to carry out basic cellular processes for growth and sustenance, most genes or pathways are actually active only under specific conditions (e.g., given time and/or tissue). Microarray has been offering a powerful tool to study gene expression patterns or active pathways under different conditions when combined with other data.

Recently, in order to identify gene modules associated to phenotypes, diseases or changing conditions, many methods have been developed by integrating interactome with gene expression data. A disease associated active module can be considered as a connected subnetwork or dysfunctional pathway in a biomolecular interaction network which has close relationship with a specific disease. Previous works to detect an active module generally include two steps. In the first step, a scoring scheme to evaluate a module's active level is adopted based on each gene's or interaction's active level from gene expression data. The scoring function is usually designed to

be an additive function of each gene's active level. Ideker et al. [9] first proposed the active pathway detection problem and gave a scoring scheme by a summational function of all genes' differentially expressed *p*-value within the subnetwork.

Since then, a hug number of progresses have been made in this direction [3, 20]. For example, Qiu et al. [20] proposed a novel network-based method to identify disease associated modules by integrating both protein-protein interaction and gene expression data in an efficient and accurate manner. The proposed regression method applied with a diffusion kernel is denoted as RegMOD, which can not only theoretically model the nonlinear effect of gene cooperativity but also run in an efficient manner. They have tested RegMOD on both simulated and real world datasets to demonstrate its efficiency. This kind of methods have also been developed for detecting diagnostic modules in diseases, such as an application in breast cancer subtype data [14].

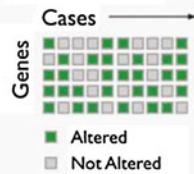
RegMOD method Based on the assumption that interacting genes have similar active scores which measure the extent to which genes respond to a specific disease. The active score of a gene is defined by a nonlinear active scoring function which considers the cooperations among genes. To estimate these underlying active scores defined by the active scoring function, a kind of observed active scores defined as differentially expressed levels of genes is calculated from the case-control microarray data. The goal is to estimate the underlying active scores for each genes which approach to the observed active scores and capture the cooperative pattern simultaneously. This is formulated as a typical regression problem by fitting the active scoring function to the observed active scores. The support vector regression method was adopted with a diffusion kernel to evaluate each gene's underlying active score. This procedure can be regarded as a smoothing process that gives each gene a new active score. It can eliminate acute changes among neighboring genes in the network and infer underlying active scores of genes whose expression level could not be measured. It can significantly improve the accuracy and robustness of the predicted activity of genes and reduce the effect of noise and incompleteness of the high-throughput data. Finally, the induced subnetworks of significantly scored genes form the active modules which are expected to be related with a specific phenotype or disease. Particularly, prioritizing genes according to their active scores can provide the order of gene's association level with a specific phenotype or disease.

In recent years, this kind of concept has been extended for other applications. As we all know, large-scale cancer genomics projects are providing an unprecedented and high-resolution view of the molecular defects in dozens of cancer types [18, 23]. Distinguishing "driver" mutations from "passenger" mutations is currently a key challenge. A further related challenge is to identify biological pathways, which are frequently perturbed within tumor cells, and lead to the acquisition of tumorigenic properties. A number of methods have been developed to address these challenges [32, 33]. Network-based methods have become a popular way to address these challenges. For example, Ciriello et al. [4] have developed a new algorithmic framework—Mutual Exclusivity Modules (MEMo)—for identifying all driver genes and altered pathways. These identified gene modules have three key properties including (i) member genes are altered more frequently than expected by chance;

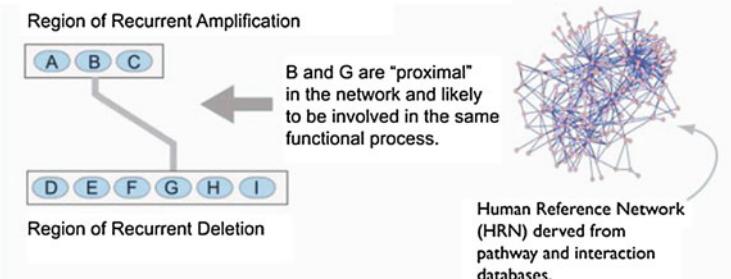
Mutual Exclusivity Modules (MEMo)

Step 1: Build Binary Event Matrix of Significantly Altered Genes

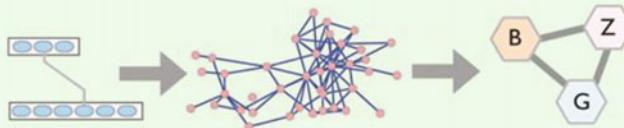
- Filter 1: Significantly Mutated Genes (SMG)
- Filter 2: Recurrently Altered Copy Number Regions of Interest (ROI)
- Filter 3: Concordant mRNA Expression



Step 2: Identify All Gene Pairs Likely to be Involved in the Same Pathway



Step 3: Build Network of Gene Pairs and Extract Cliques



Step 4: Assess Each Clique for Mutual Exclusivity

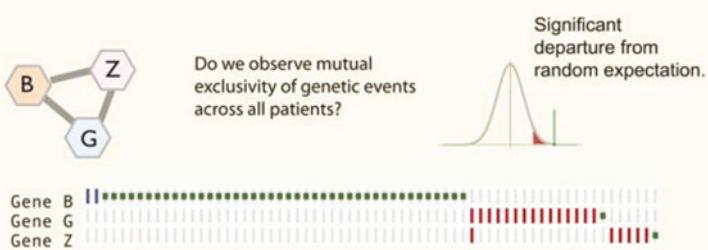


Fig. 18.2 Overview of MEMo. The figure is adapted from [4]

(ii) member genes are likely to participate in the same biological pathway or process; and (iii) genomic events within the network exhibit a statistically significant level of mutual exclusivity.

MEMo method MEMo detects sets of interacting genes that are recurrently altered, and exhibit patterns of mutually exclusive genetic alteration across multiple patients. The algorithm consists of four key steps (Fig. 18.2). In Step 1, the algorithm uses the mutation events across all samples with multiple constraints to generate a binary event matrix of all related genes. Step 1 results in a set of recurrently altered entities and leads to a binary matrix M , where each entry m_{ij} refers to the status of gene i in the sample j . In Step 2, MEMo determines all pairs of genes that are functionally connected to one another, based on prior pathway and network knowledge. In Step 3, MEMo builds a graph of all similar gene pairs, by creating an edge between two genes if they are found similar by the network proximity metric in Step 2. MEMo then extracts from this graph all maximal cliques. These cliques represent local clusters, containing functionally coherent proteins. In Step 4, MEMo determines the statistical significance of each clique identified in Step 3 whether it exhibits mutual exclusivity characteristics.

18.4 Data Integration by Network Aggregation

Large-scale cancer genomics projects like The Cancer Genome Atlas (TCGA) have already produced various types of genomic data for over 20 cancers from thousands of patients [18, 23]. These rapidly evolving projects are collecting a huge number of multiple and diverse genome-scale data sets to address clinical and biological questions. The availability of such huge data enables new challenges for data integration to capture the heterogeneity of biological processes and phenotypes [13, 29, 31]. These challenges include the relative small number of samples compared to the large number of measurements, the diversity in scale, bias and noise in each data set, and the consistency and complementary nature of the information provided by different types of data. Network-based integration method has played important roles for addressing all of these challenges together [29].

Recently, Wang et al. [22] proposed a similarity network fusion (SNF) approach which uses networks of samples as a basis for integration. SNF creates a patient network for each data and then combine them in an stimulative aggregation way. The SNF method consists of two main steps (Fig. 18.3). The first step is to construct a sample-similarity network for each data type. The second step is to integrate these networks into a single similarity network using a nonlinear combination method. The combined network can capture both shared and complementary information from different data sources. SNF is not sensitive to the sample number which enable it derive useful information even from a small number of samples and combine data types of different sample sizes. By applying SNF to five different human cancers, they demonstrated that SNF yields coherent, clinically relevant patient subtypes and improves on the performance of popular integrative approaches.

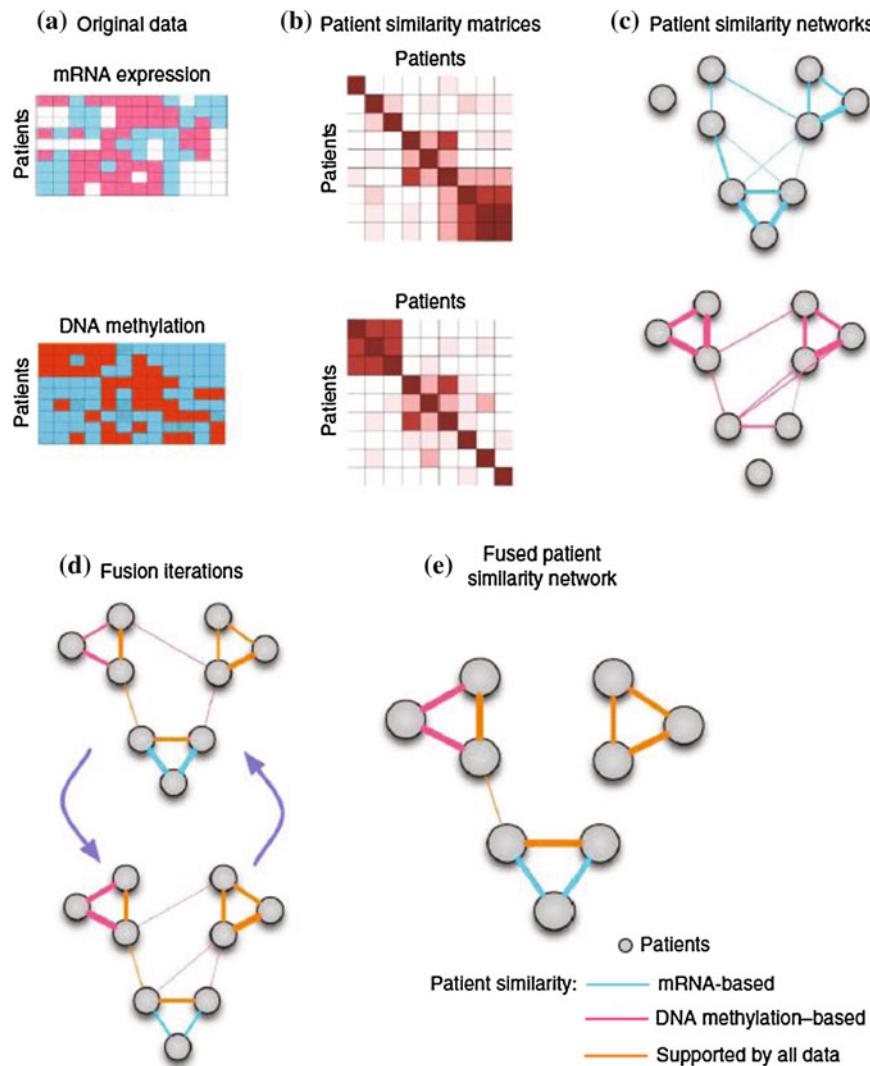


Fig. 18.3 Overview of SNF. **a** Toy example of mRNA expression and DNA methylation data sets for the same cohort of patients. **b** The corresponding patient-by-patient similarity matrices for each data type. **c** Patient-by-patient similarity networks from the patient-by-patient data with nodes corresponding to patients and edges corresponding to patients-patient similarities. **d** SNF iteratively updates each of the networks by considering the information of the other networks. **e** The iterative network fusion results in convergence to the final fused network. This figure is adapted from [22]

SNF method Let's denote a patient similarity network as a graph $G = (V, E)$, where the vertices V correspond to the patients x_1, x_2, \dots, x_n and the edges E are weighted by how similar the patients are. The edge weights reflected by the similarity matrix W with $W(i, j)$ indicating the similarity between patients x_i and x_j :

$$W(i, j) = \exp\left(-\frac{\rho^2(x_i, x_j)}{\mu \varepsilon_{ij}}\right),$$

where $\rho(x_i, x_j)$ is the Euclidean distance between patients x_i and x_j , where μ is a hyperparameter and ε_{ij} is used to control the scaling problem.

To compute the fused matrix from multiple types of measurements, a full and sparse kernel on the vertex set V is defined as a normalized weight matrix $P = D^{-1}W$, where D is the diagonal matrix whose entries $D(i, i) = \sum_j W(i, j)$, so that $\sum_j P(i, j) = 1$. Empirically, they have adopted a adapted normalization [22]. They further define K nearest neighbors (KNN) to measure local affinity $S(i, j)$. We take the case with two data types as an example, after calculating the status matrices $P^{(1)}$ and $P^{(2)}$ from two input similarity matrices, and the kernel matrices $S^{(1)}$ and $S^{(2)}$. Given the initial two status matrices as $P_{t=0}^{(1)} = P^{(1)}$ and $P_{t=0}^{(2)} = P^{(2)}$ at $t = 0$. The key of SNF is to iteratively update similarity matrix as follows:

$$P_{t+1}^{(1)} = S^{(1)} P_t^{(2)} (S^{(1)})^T,$$

$$P_{t+1}^{(2)} = S^{(2)} P_t^{(1)} (S^{(2)})^T,$$

This procedure updates the status matrices which generate two parallel interchanging diffusion processes. The overall status matrix is computed after termination as

$$P_c^{(c)} = \frac{P_c^{(1)} + P_c^{(2)}}{2}.$$

18.5 Data Integration by Network-Regularized Methods

To achieve data integration, Zhang et al. [29] proposed a computational framework SNMNMF for reconstructing miRNA regulatory modules based on the integration of multiple genomic data sources. Three types of data are used: predicted miRNA-gene interactions, the expression profiles of miRNAs and genes, and the gene-gene interaction network constructed based on protein-protein interaction and DNA-protein interaction networks. The predicted miRNA-gene targets serve as a static superset, while the dynamic expression profiles of miRNAs and genes are used to identify target relationships that are concurrently active. This signal is enhanced by the protein-protein interactions, since the ultimate effect of miRNA regulation is to regulate protein activities. In order to integrate the three information sources, Zhang et al. [29] propose a novel and efficient machine learning technique. The method integrates miRNA and gene expression profiles in a framework of multiple nonnegative matrix factorization, and simultaneously integrates networked data in a regularized manner. To enhance the signal-noise separation and improve the interpretability of the modules, we look for sparse solutions of the membership functions by applying sparsity penalties.

A theoretical derivation shows that the learning and optimization model can be effectively solved by an iterative algorithm. This method has been applied to a data set of human miRNA and gene expression profiles (from TCGA ovarian cancer samples), a miRNA-gene interaction network, and a gene interaction network to demonstrate its efficiency.

SNMNMF method Here we describe this framework for the simultaneous integration of multiple data types to identify miRNA-gene co-modules (Fig. 18.4). To identify miRNA-gene co-modules, they designed an objective function with three components. The first is based on the non-negative miRNA and gene expression matrices X_1 and X_2 . The second considers the effects of gene-gene interactions. The last considers the effects of predicted miRNA-gene interactions. By optimizing this objective function, they obtain a joint decomposition of X_1 and X_2 that together reveals miRNA-gene regulatory modules inherent in the expression data and satisfies constraints based on prior information.

They assume that there is a common basis matrix W for the miRNA and gene expression matrices X_1 and X_2 . The two expression matrices have dimensions $s \times m$ and $s \times n$, respectively, and will be factored into W and two coefficient matrices H_1 and H_2 . This representation of the expression data can be derived by optimizing the following objective function:

$$F_1(W, H_1, H_2) = \sum_{I=1,2} \|X_I - WH_I\|_F^2.$$

where H_1 and H_2 have dimensions $k \times m$ and $k \times n$, respectively. The parameter k is chosen prior to optimization. The solution is often not unique, and may be sensitive to noise in the expression data. Both of these limitations may confound the module discovery process. For these reasons, they consider to improve it by incorporating prior knowledge into the objective function. The prior knowledge consists of predicted miRNA-gene interactions and gene-gene interactions. The essence of this learning method is to define constraints for the co-module identification framework such that any variables linked in these two datasets are more likely to be placed into the same co-module. In addition to improving the biological relevance of the results, such constraints can greatly facilitate the discovery of co-modules by narrowing down the large search space.

Let A denote the adjacency matrix of a gene interaction network, and B denote the adjacency matrix of a bipartite miRNA-gene network. They enforce “must-link” constraints by maximizing the following objective function:

$$O_1 = \sum_{ij} a_{ij} (h_i^2)^T h_j^2 = \text{Tr}(H_2 A H_2^T).$$

This term ensures that genes with known interactions have similar coefficient profiles. Similarly, the interactions between genes and miRNAs can be encoded by the following objective function:

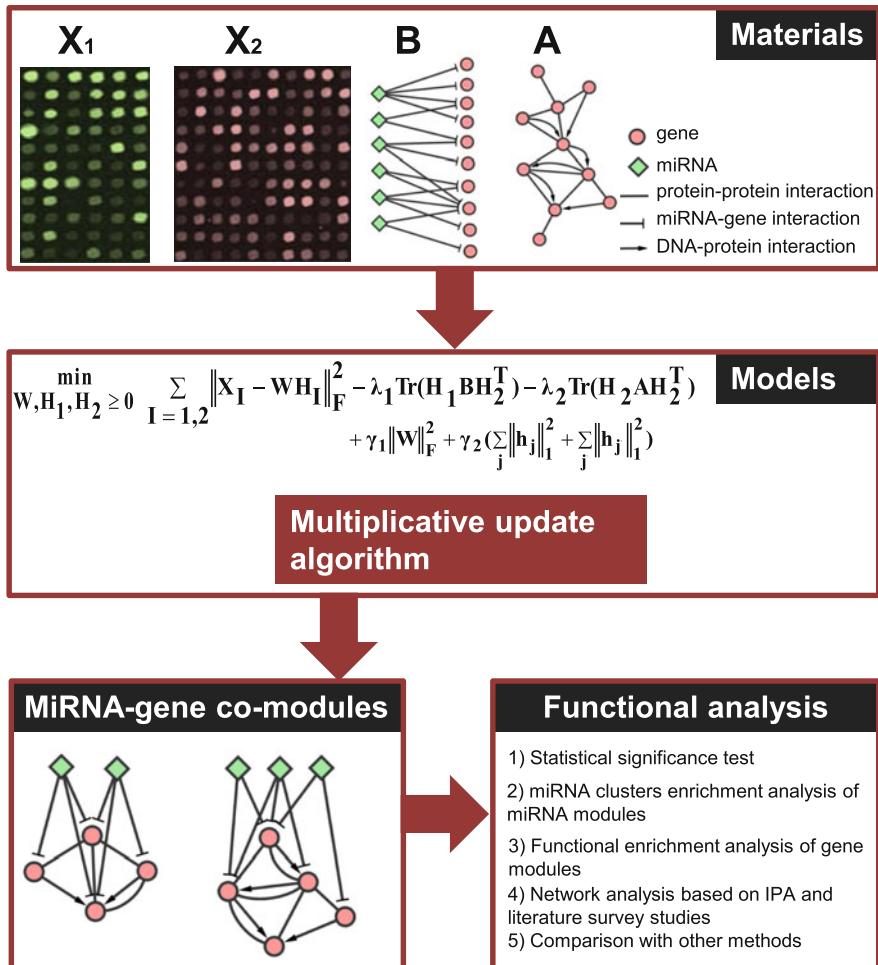


Fig. 18.4 Overview of SNMNMF for identifying miRNA-gene co-modules. The inputs are (1) two sets of expression profiles (X_1 and X_2) for miRNAs and genes across the same set of samples; (2) a gene-gene interaction network (A), including protein-protein interactions and DNA-protein interactions; and (3) a list of predicted miRNA-gene regulatory interactions (B) based on sequence data. SNMNMF simultaneously factor the miRNA and gene expression matrices into a common basis W and two coefficient matrices H_1 and H_2 . At the same time, additional knowledge is incorporated into this framework with network-regularized constraints. Sparsity constraints are also imposed on this framework so as to obtain easily interpretable solutions. The decomposed matrix components provide information about miRNA-gene regulatory co-modules. Then the co-modules are identified based on shared components (a column in W) with significant association values in the corresponding rows of H_1 and H_2 . This figure is produced from [29]

$$O_2 = \sum_{ij} b_{ij} (h_i^1)^T h_j^2 = Tr(H_1 B H_2^T).$$

The inputs are the miRNA and gene expression matrices X_1 and X_2 with dimensions $s \times m$ and $s \times n$, respectively, an $m \times n$ matrix B of predicted miRNA-target interactions, and an $n \times n$ gene-gene interaction network A . To discover miRNA-gene regulatory co-modules, this method combines the three objectives defined in the previous sections into a single optimization function.

The parameters λ_1 and λ_2 are weights for the must-link constraints defined in A and B . The first term favors modules with miRNA and gene expression profiles that are correlated in the common basis matrix W . The second term, $Tr(H_2 A H_2^T)$, summarizes all the must-link constraints in the gene-gene network. The third term, $Tr(H_1 B_{12} H_2^T)$, summarizes all the must-link constraints in the miRNA-gene network.

In this Network-Regularized Multiple NMF (NMNMF) framework, a strategy to make the coefficient matrices H_1 and H_2 sparse has been adopted. The SNMNMF is formulated as follows:

$$\begin{aligned} F(W, H_1, H_2) = & \sum_{I=1,2} \|X_I - WH_I\|_F^2 - \lambda_1 Tr(H_2 A H_2^T) - \lambda_2 Tr(H_1 B H_2^T) \\ & + \gamma_1 \|W\|_F^2 + \gamma_2 \left(\sum_j \|h_j\|_1^2 + \sum_{j'} \|h_{j'}\|_1^2 \right) \end{aligned}$$

where h_j and $h_{j'}$ are the j th and j' th columns of H_1 and H_2 respectively. The term $\gamma_1 \|W\|_F^2$ limits the growth of W , while $\gamma_2 (\sum_j \|h_j\|_1^2 + \sum_{j'} \|h_{j'}\|_1^2)$ encourages sparsity.

In the basic NMF problem, the objective function is not convex in W , H_1 and H_2 . Therefore, it is unrealistic to expect a standard optimization algorithm to find the global minimum. Zhang et al. [29] have developed an algorithm that efficiently converges to a local minimum by iteratively updating the matrix decomposition. Under the rules laid out below, the objective function $F(\cdot)$ is guaranteed not to increase when the decomposition is updated. Furthermore, the objective function remains invariant if and only if W , H_1 and H_2 are at a stationary point. This behavior can be proved in the same way as for the classical NMF algorithm. We note that H_1 and H_2 are updated at the same time based on their current values at each iteration. The time complexity of the proposed algorithm is $O(tk(s+m+n)^2)$, where t is the number of iterations.

18.6 Module Detection in Multi-layer Networks

Modularity has been considered to be one major organization principle of biological system. Biological modules as a critical level of biological hierarchy and relatively independent units play key roles in the form, organization and function of biological

networks [27]. How to determine the modular structure of biological networks is an essential step for understanding biosystems. In the past decade, a huge number of computational methods have been developed for detecting network modules and analyzing the network structure of biological networks [25, 26, 28].

One popular class of methods for dissecting modular structure is based on optimizing a quality function called modularity to partition the network into modules. However, it has recently been shown that the resolution of the modularity based methods is intrinsically limited. Li et al. [11] proposed a new quality function called modularity density to conquer the resolution limit problem of modularity. Moreover, Zhang et al. [28] has introduced this new measure into the modular analysis of biomolecular networks and develop new algorithms including a simulated annealing approach and a spectral k -means method for detecting functional modules in protein-protein interaction networks.

Moreover, network alignment is a general problem for finding the conserved subnetworks or functional modules in two or more species [12]. Different with this framework, Yan et al. [24] developed a novel network-based framework OrthoClust for clustering data across multiple species (Fig. 18.5). OrthoClust extends the idea of functional modules into a cross-species manner by integrating the networks of individual species using orthology relationships of genes between species. The key of OrthoClust is using an objective function to describe the modules across species. Next a simulated annealing approach was used to solve this optimization problem. They have demonstrated the efficiency of this method by applying it to worm and fly co-expression networks generated by the modENCODE consortium [6].

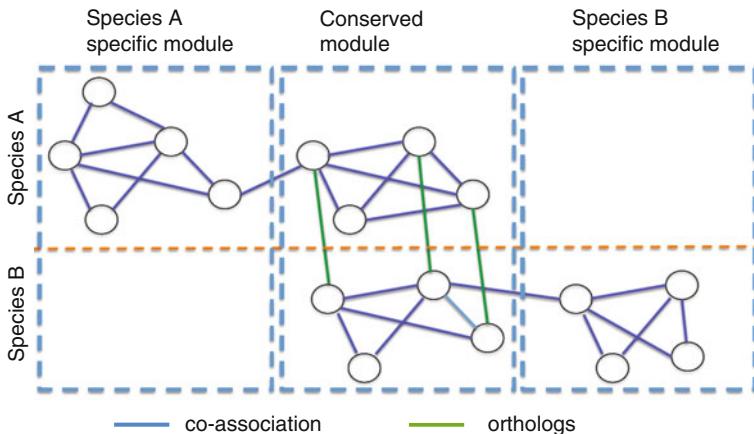


Fig. 18.5 Module illustration in a toy multi-layer network. A multi-layer network is composed of the individual networks of species A and B and the orthologous links between genes connecting these two networks. The three modules are the one conserved with genes from both species, corresponding to fundamental biological functions across different species. The left and right ones are specific modules consisting of genes from species A and B, respectively. They correspond to novel functions that emerged in each of the two species. The figure is adapted from [24]

OrthoClust method OrthoClust first defines a objective function to describe and detect modules in a multi-layer network. Specifically, every node is assumed to take a discrete label σ ranging from 1 to q . All nodes with the same label will be assigned to the same module and q is predefined maximum number of modules. For two-layer networks each label assignment or configuration is characterized by an objective function defined as:

$$H = - \left(\sum_{i,j \in S_1} \Lambda_{ij}^1 \delta_{\sigma_i \sigma_j} + \sum_{i,j \in S_2} \Lambda_{ij}^2 \delta_{\sigma_i \sigma_j} + \kappa \sum_{(i,j') \in (S_1, S_2)} w_{ij'} \delta_{\sigma_i \sigma_{j'}} \right),$$

where S_1 and S_2 are the gene sets for the two species, respectively; $\Lambda_{ij} = A_{ij} - k_i k_j / 2m$, with $k_i = \sum_j A_{ij}$, $m = \sum_i k_i / 2$, and A is a network adjacency matrix. The superscripts (1 or 2) correspond to the networks of two species. The value of the Kronecker delta $\delta_{\sigma_i \sigma_j}$ equals one if nodes i and j have the same label and zero otherwise. The first two terms of the cost function H are essentially the modularity functions of two individual networks. OrthoClust further employs a standard simulated annealing procedure to solve this problem.

More generally, the community structure in two or more slices of a series of time-varying networks has only been addressed by a few literatures [16]. For example, Zhang et al. [31] proposed the concept of common community structure in two or more networks of a series of time-varying networks. The basic assumption is that essential and common community structure may exist in two or more networks, and local dynamic changes may happen. This is very realistic in time-varying networks of many robust systems.

They initially focused on the problem in two networks, which is a simple but useful case in analyzing time-varying networks. To formulate the problem easily, they considered the common notation of clustering or community structure detection problems. The objective of classical community detection in networks is to partition the vertex set V of the graph $G(V, E)$ with $|V| = N$ into K distinct subsets in a way that densely connected groups of vertices are placed in the same community. In this case, a convenient representation of a given partition is the partition matrix $U = [u_{ik}]$ (or $[u_i]$, u_i is a membership vector) with size of $N \times K$. And $u_{ik} = 1$ if and only if vertex i belongs to the k th subset in the partition, otherwise it is zero. From the definition of the partition, it clearly follows that $\sum_{k=1}^K u_{ik} = 1$ for all i . The generalization of the hard partition follows by allowing u_{ik} to attain any real value from the interval $[0, 1]$, and the corresponding matrix is also called membership matrix.

They adopted the popular membership matrix representation to formulate the problem. A basic consideration is that an edge between vertex v_1 and v_2 implies the similarity of v_1 and v_2 , and likewise, the absence of an edge implies dissimilarity, i.e., $a_{ij} \simeq u_i u_j^T$ or $A \simeq UU^T$, where $A = (a_{ij})$ is the adjacency matrix of a network. At the same time, the same vertices in two networks should have similar membership vectors. These considerations can be formulated as:

$$\begin{aligned}
\min \quad & \sum_{g=1}^2 \|A_g - H_g H_g^T\|_F^2 + \lambda_1 \sum_{g=1}^2 \|H_g - H\|_1 + \lambda_2 \|H\|_1 \\
\text{s.t.} \quad & \sum_{k=1}^K (H_g)_{ik} = 1; \quad (H_g)_{ik}, H_{ik} \geq 0; \\
& g = 1, 2; \quad i = 1, \dots, N; \quad k = 1, \dots, K.
\end{aligned}$$

where A_g is the adjacency matrix of network $G(V, E_g)$, H_g is the membership matrix of network $G(V, E_g)$, $\|\cdot\|_F$ and $\|\cdot\|_1$ are the entrywise matrix norm ($\|\cdot\|_F$ is known as the Frobenius norm). We note that H is the virtual membership matrix which reflects the membership of nodes determined by the topological information of two networks. By solving this problem, one can discover the communities of two networks as well as their common communities.

18.7 Disease-Aging Network Analysis

One of the challenging problems in biology and medicine is to explore the underlying mechanisms of genetic diseases. During the last decades, great efforts have been devoted to identifying disease-related genes and disease-related pathways. Progresses have been achieved both in understanding the mechanisms of specific diseases and in identifying key proteins as potential drug targets. However, these single gene-based methods are far from enough in elucidating complex diseases. For example, Alzheimer disease, a kind of neurological disease, is related with at least 12 genes (Online Mendelian Inheritance in Man, OMIM). The mechanism of this kind of heterogeneity diseases cannot be totally uncovered by the conventional gene-by-gene or pathway-by-pathway methods because most cellular components exert their functions through complicated networks of signal transductions, gene regulations, metabolic reactions, and protein interactions.

Network-based methods to study human genetic diseases appear along with the concept of ‘omics’ and the growth of high-throughput data. For example, Jonsson and Bates studied the global topological features of cancer proteins in a predicted human protein-protein interaction (PPI) network. In their work, features of diseases were uncovered from a global analysis, but they did not consider the effect of essential genes. Combining with essential genes, Goh et al. [7] found some different conclusions in a human disease network.

Research on aging is helpful to understand the nature of diseases by integrating disease and aging information at a network level. Aging is another complex process in addition to genetic diseases controlled by both environmental and genetic factors. Researchers began to investigate aging process at a systems level by employing protein networks. For example, Budovsky et al. compiled a complete list of longevity genes from different species, mapped them to 211 orthologs in human, and constructed a human longevity network using protein-protein interactions. Wang et al.

[21] highlighted the intricate relationships between aging and diseases since the process of aging is a gradual decay of homeostatic mechanisms affecting our susceptibility to disease and our ability to recover from illness and other stressors. We note that their relationships have been pointed out for a long time, but seldom been investigated from the systems perspective. Recently, some progresses are reported. Budovsky et al. verified the existence of evolutionary and molecular links between longevity and cancer. Wolfson et al. highlighted the importance of some pathways by combining the network of human age-related disease proteins and longevity-associated proteins, especially through those hubs involved in the crossroad of longevity and age-related disease network.

At the same time, there is a pressing need to associate genetic diseases and aging at a network level. Firstly, only a small number of well known age-associated diseases have been considered, and thousands of different kinds of genetic diseases remain untouched. Secondly, longevity genes are actually not equal to aging genes. Longevity genes are alleles that have been observed to have higher frequency in centenarian than others. Different from longevity genes, aging genes are those genes that have been identified in human or animal models to have the ability to change the aging process as a whole, or at least to a large degree. Combining genes that are related to aging process with diseases may reveal the nature of complex diseases. Thirdly the problem how close the genetic diseases and aging process are and why they are close to each other have not been solved until now.

Wang et al. [21] analyzed the relationships between aging and disease genes by integrating human PPI, known disease-gene associations and known aging-gene associations into a disease-aging network (DAN), then classified diseases genes based on the derived network, and further quantified the contribution of aging genes to association between each pair of diseases. Specifically, they firstly constructed a disease-aging network (DAN) and analyzed its topological properties. Then they identified the relationship between aging genes and disease genes, and categorized diseases into two types: type I disease genes are significantly close to aging genes, but type II disease genes are not. Furthermore, they examined the features of topology and structure for the disease-aging network from a systems perspective. Theoretical results showed that type I diseases were in a central position of a PPI network while type II were not. Moreover, they defined an asymmetric closeness based on PPI network to describe close associations between diseases, and found that aging genes make a significant contribution to most of disease associations comparing with genes having the same number of links.

The disease-aging network They constructed a network of aging and genetic diseases named disease-aging network (DAN), which is a connected PPI network whose nodes are known aging and disease genes (Fig. 18.6a). According to OMIM and GenAge, there are 1,438 genes related to aging or diseases in addition. They mapped all these genes to nodes in the PPI network of Human Protein Reference Database (HPRD), and then extracted the maximum connected component as DAN. As shown in Fig. 18.6a, aging genes are marked by nodes with black border while disease genes are colored according to their categories of diseases, which is a curated classification of all OMIM diseases. If one gene is reported to be related with more

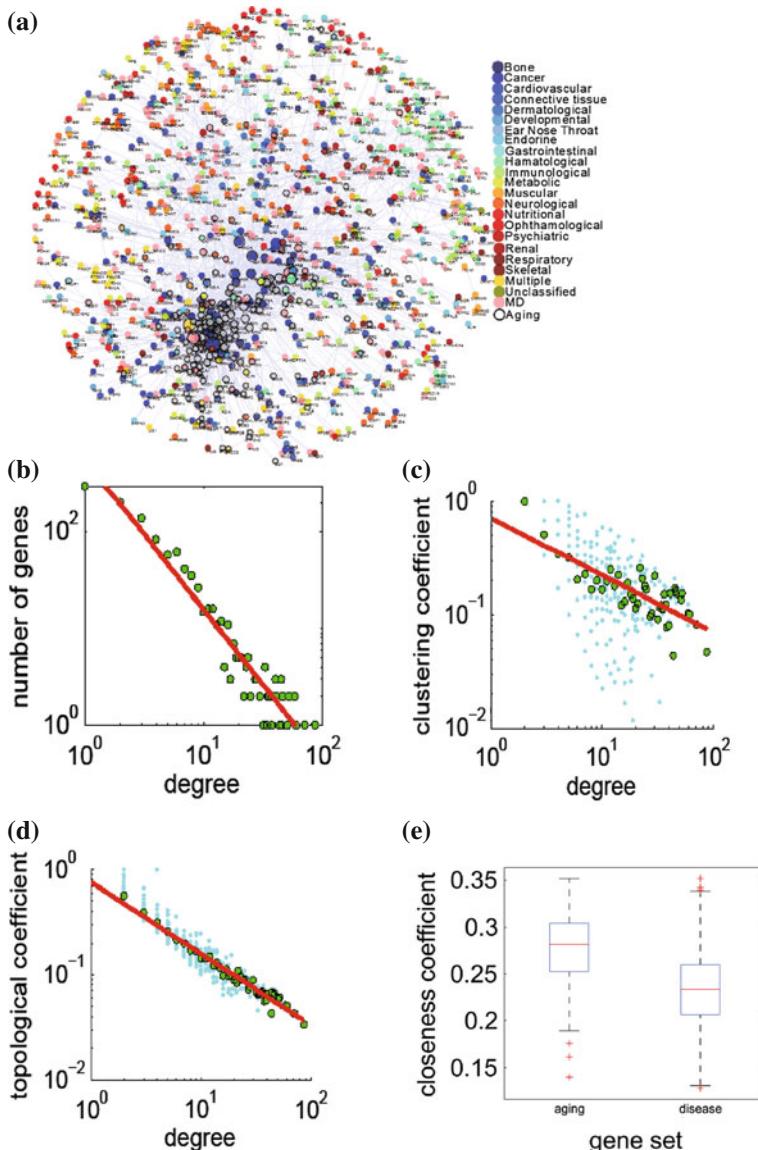


Fig. 18.6 The disease-aging network (DAN) and its topological properties. **a** A protein-protein interaction network connecting aging and disease. Non-disease aging genes are colored in grey and disease genes are colored by their types. MD in the figure means that the genes are involved in multiple gene sets. Refer to Materials and Methods for detailed information about aging genes and classification of disease genes. **b, c** and **d** Basic network features of disease aging network. Refer to Materials and Methods for detailed information about definition of network features. **e** Box plot for closeness centrality of disease and aging genes in DAN. The figure is adapted from [21]

than one category, it will be colored in pink (labeled as MD in Fig. 18.6a). The size of nodes and the color of edges correspond to the degree and betweenness centrality, respectively.

They also showed that this network is scale-free (Fig. 18.5, which shows an unusual degree of robustness. In the disease-aging network, average degree of nodes with black borders is 14.3, which is significantly larger than that of disease genes 4.9 with a p -value = 8.4e-36 (Wilcoxon rank sum test), implying the importance of aging genes in this network's connectivity. They further calculated the clustering coefficient of each node in the network and showed that clustering coefficient in DAN decreases with the increase of nodes' degree, indicating that DAN has a hierarchical structure (Fig. 18.6c). In a hierarchical network, a high degree hub connects some local communities, suggesting that the network has two levels of organization. The topological coefficient decreases with the number of links, which clearly shows that, disease and aging hub genes do not have more common neighbors than genes with fewer links (Fig. 18.6d). This fact indicates that the hubs may not locate together in a few densely connect modules like cliques in DAN. Aging genes (nodes with black borders) tend to locate in the central part of DAN. To measure 'central' quantitatively, they calculated closeness centrality and showed that the average closeness centrality value of aging genes is much greater than that of disease genes (p -value < 5e-40) (Fig. 18.6e). More detailed analysis can be found in [21].

18.8 Systematic Analysis of Human Transcription Factor Networks

In the human genome, genes demonstrate dramatic diversity in expression level in different tissues and/or conditions, controlled by a number of transcription factors (TFs). The TFs can work cooperatively to facilitate or inhibit the recruitment of RNA polymerase and display high specificity in ligand activation thereby precisely controlling the condition-dependent expression of target genes. The mutual interactions among TFs and their activity shape the major features of cellular identity and complex functions, which makes the study of transcriptional combinatorial regulation of vital importance.

The rapid development of biological technologies such as whole-genome sequencing, large-scale expression profiling and chromatin immunoprecipitation enables it possible to understand the combinatorial transcriptional regulation of TFs from different points of view and explore the cell or condition specificity of TF regulation. Gene expression profiles, protein interactions, and even genome sequence have been employed to discover combinatorial TF interactions in diverse view. Recently, the ENCODE Consortium provides comprehensive high-throughput ChIP-seq data for reliable TF-binding interactions [5].

However, comprehensive generation of regulatory networks for biological systems in different tissues or conditions has been limited in many aspects especially that the key elements only collected from individual experiments targeting one cell type and

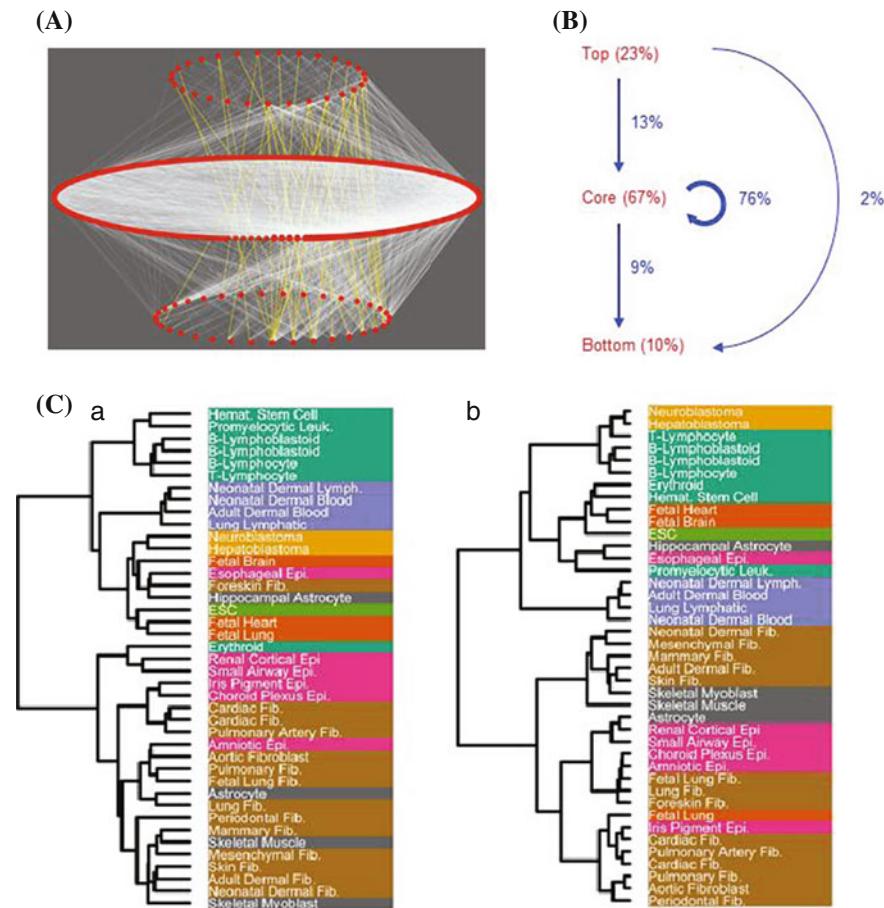


Fig. 18.7 Hierarchical structures in the 41 regulatory networks. **A** ESC regulatory network is arranged into *Top*, *Core* and *Bottom* layers. **B** Distributions of nodes (red) and interactions (blue) among three layers *Top*, *Core* and *Bottom* in the hierarchical layouts of 41 regulatory networks. **C** The hierarchical clustering of 41 cell types, where the color indicates which classes they belong to. (a) The clustering reported in [17], which is based on the pairwise Euclidean distances between the NND vectors of the corresponding TF regulatory networks ($RI = 0.801$). (b) Our clustering, which is based on the distribution of the downstream targets of the seven signal transducer and activator of transcription (STAT) proteins ($RI = 0.856$). The figure is adapted from [32]

one TF at a time. The general correlation-based method using gene expression data cannot well capture the directionality which is a key feature of regulatory networks. Thus, only a limited number of regulatory networks have been constructed which limited the systematic analysis of the cooperativity of TFs among different conditions and/or tissues. More recently, the DNaseI footprinting technique was adopted to simultaneously determine the sequence-specific regulatory networks to study the human cellular diversity within 41 different cell types. This technique provides an

unprecedented opportunity to the dynamics of combinatorial TF interactions. It opens a door to learn the core structure of human regulatory networks and their specific component subnetworks by comparative analysis.

Neph et al. [17] have mapped comprehensive complex networks of regulatory interactions between human TFs in 41 cell types and explored their dynamics in different aspects, which provides an unprecedented valuable resource for further deciphering the regulatory mechanism. In such a regulatory network, the nodes are the TFs and the edges are the cross-regulation of one TF by another. The preliminary extensive analysis on such human regulatory networks comprising of connections among 475 sequence-specific TFs of 41 diverse cell and tissue types demonstrates interesting characteristics on the network topology and dynamics.

Zhang et al. [32] revisited the 41 regulatory networks and found that the topological characteristics of just a few (7) of TFs and their targets can distinguish the cell identities accurately, indicating significant local TF regulation difference among cells. More interestingly, the regulatory networks of lineage-related cell types were shown to share a significant common global regulatory system. Furthermore, these regulatory networks reveal significant hierarchical structures. By employing the inherent cell-type diversity of these regulatory networks, they identified house-keeping regulatory interactions and cell type-specific interactions. They applied the identified cell type-specific interactions coupled with known protein complexes data to discover regulatory machinery. Together, our analysis from global hierarchical organization to local interactions reveals a systematic understanding of function and organizing principles of the human regulatory networks.

Hierarchical structures of 41 cell specific regulatory networks They studied hierarchical organization of regulatory networks of the 41 cell types. For each network, the vertex-sort algorithm [10] identified the Top, Core, and Bottom layers (Fig. 18.7a). The percentages of TFs in the three layers in the 41 regulatory networks were reported in [32]. On average, 67 % of TFs are classified into the Core layer, 23 % into the Top layer, and the least amount 10 % of TFs into the Bottom layer (Fig. 18.7b). This distribution is different from that of the Yeast regulatory network, where the Core and Bottom layers have similar amounts of TFs (43 vs 40 %), whereas the Top layer has lesser TFs (17 %). Interestingly, the regulatory network of ESC has a significantly low number of TFs in the Top layer (6 %, p value < 0.001) and a significantly high number of TFs in the Core layer (85 %, p value < 0.001) when compared to regulatory networks of other cell types. While the Bottom layer of the ESC network has a size similar to that of the other networks. Interestingly, the similarity of 41 regulatory networks in terms of the Top layers characterizes the corresponding cell types well, whereas the Core layers and the Bottom layers fail to distinguish the cell types well.

The distributions of interactions among three layers of 41 regulatory networks were examined. 76 % of interactions are distributed between TFs within the Core layer (Fig. 18.7B). Both the high percentages of TFs and their interactions among them in the Core layer reveal the complex regulatory relationships between TFs in the cell type-specific networks. The distribution suggests that TFs in the Top layer mainly regulate TFs in the Core layer.

Topological features of a few TFs can define cell identity accurately Neph et al. [17] clustered all cell-type networks using the so-called NND vectors which is a global measure defined based on the normalized degree of each node. The resulting network clustering strikingly groups anatomical and functional cell-type groupings into clearly pre-annotated groups (Fig. 18.7A). This demonstrates the global topological properties of networks characterize cell identity well. Strikingly, Zhang et al. [32] showed that even the local structures of the networks can distinguish the anatomical and functional identity of 41 cell and tissue types. For example, when using just 7 STAT proteins and information on how these proteins connect with their targets to classify the 41 cell types, they showed even better classification result (Fig. 18.7B, RI = 0.856 vs RI = 0.801). This suggested the importance of classification of cell types using a small number of randomly selected TFs.

Surprisingly, a random set of 5 to 9 TFs gives a comparable classification (average RI lies within the range from 0.7 to 0.9). This suggests that local topological structures of regulatory networks are significantly specific to the cell types. In fact, a closer examination finds that the TFs regulated by the STATs are annotated with different GO terms in different regulatory networks. TFs that are regulated by STATs in the ESC, but not in the Hematopoietic stem cell, are enriched in GO:0045165 (cell fate commitment, Benjamini corrected p value = 2.72e-7). On the other hand, TFs that are regulated by STATs in the Hematopoietic stem cell, but not in the ESC, are enriched in GO:0048534 (hemopoietic or lymphoid organ development, Benjamini corrected p value = 0.03).

18.9 Conclusion

Network has been well recognized as an valuable tool to explore the interactions of individual components which are ultimately responsible for an organisms' form, organization and functions. The huge accumulation of various omics data greatly enriches the field of biological molecular networks. Naturally, network-based analysis of cellular system can lead to deep insights into biological systems by discovering biological functions and revealing essential mechanisms at the molecular level in a system-wide manner. Moreover, the paradigm has expanded to non-molecular biological networks such as patient-patient network and so on.

Here, we briefly surveyed the related topics in this field with the special emphasis on the network-based data integration studies recently. As a rapid ongoing field, further research works and efforts from both experimental and theoretical perspectives are expected to exploit the great potential on understanding fundamental mechanism of living organisms and challenging both biological and medical problems not at individual component level but at a system-wide level. Limited by the space, several related studies on molecular networks were not included in this survey.

Acknowledgments This work was supported by the National Natural Science Foundation of China, No. 61379092, 61422309 and 11131009, the Strategic Priority Research Program of the Chinese Academy of Sciences (CAS) (XDB13040600), the Foundation for Members of Youth Innovation Promotion Association, CAS, The Outstanding Young Scientist Program of CAS, and the Key Laboratory of Random Complex Structures and Data Science, CAS.

References

1. Albert, R.: Scale-free networks in cell biology. *J. Cell. Sci.* **118**, 4947–4957 (2005)
2. Barabási, A.L., Oltvai, Z.N.: Network biology: understanding the cell's functional organization. *Nat. Rev. Genet.* **5**(2), 101–113 (2004)
3. Chuang, H.Y., Lee, E., Liu, Y.T., Lee, D., Ideker, T.: Network-based classification of breast cancer metastasis. *Mol. Syst. Biol.* **3**, 140 (2007)
4. Ciriello, G., Cerami, E., Sander, C., Schultz, N.: Mutual exclusivity analysis identifies oncogenic network modules. *Genome Res.* **22**, 398–406 (2012)
5. ENCODE Project Consortium: An integrated encyclopedia of DNA elements in the human genome. *Nature* **489**(7414), 57–74 (2012)
6. Gerstein, M.B., Lu, Z.J., Van Nostrand, E.L., Cheng, C., Arshinoff, B.I., Liu, T., Yip, K.Y., Robilotto, R., Rechtsteiner, A., Ikegami, K., et al.: Integrative analysis of the *Caenorhabditis elegans* genome by the modENCODE project. *Science* **330**(6012), 1775–1787 (2010)
7. Goh, K.I., Cusick, M.E., Valle, D., Childs, B., Vidal, M., Barabsi, A.L.: The human disease network. *Proc. Natl. Acad. Sci. USA* **104**(21), 8685–8690 (2007)
8. Hofree, M., Shen, J.P., Carter, H., Gross, A., Ideker, T.: Network-based stratification of tumor mutations. *Nat. Methods* **10**, 1108–1115 (2013)
9. Ideker, T., Ozier, O., Schwikowski, B., Siegel, A.F.: Discovering regulatory and signalling circuits in molecular interaction networks. *Bioinformatics* **S1**, S233–240 (2002)
10. Jothi, R., Balaji, S., Wuster, A., Grochow, J.A., Gsponer, J., Przytycka, T.M., Aravind, L., Babu, M.M.: Genomic analysis reveals a tight link between transcription factor dynamics and regulatory network architecture. *Mol. Syst. Biol.* **5**, 294 (2009)
11. Li, Z., Zhang, S., Wang, R., Zhang, X.S., Chen, L.: Quantitative function for community detection. *Phys. Rev. E* **77**, 36109 (2008)
12. Li, Z., Zhang, S., Wang, Y., Zhang, X.S., Chen, L.: Alignment of molecular networks by integer quadratic programming. *Bioinformatics* **23**, 1631–1639 (2007)
13. Li, W., Zhang, S., Liu, C.C., Zhou, X.J.: Identifying multi-layer gene regulatory modules from multi-dimensional genomic data. *Bioinformatics* **28**, 2458–2466 (2012)
14. Liu, Z., Zhang, X.S., Zhang, S.: Breast tumor subgroups reveal diverse clinical prognostic power. *Sci. Rep.* **4**, 4002 (2014)
15. Liu, Z., Zhang, S.: Toward a systematic understanding of cancers: a survey of the pan-cancer study. *Front Genet.* **5**, 194 (2014)
16. Mucha, P.J., Richardson, T., Macon, K., Porter, M.A., Onnela, J.P.: Community structure in time-dependent, multiscale, and multiplex networks. *Science* **328**(5980), 876–878 (2010)
17. Neph, S., Stergachis, A.B., Reynolds, A., Sandstrom, R., Borenstein, E., Stamatoyannopoulos, J.A.: Circuitry and dynamics of human transcription factor regulatory networks. *Cell* **150**, 1274–1286 (2012)
18. Cancer Genome Atlas Network: Comprehensive molecular portraits of human breast tumours. *Nature* **490**, 61–70 (2012)
19. Newman, M.E.J.: The structure and function of complex networks. *SIAM Rev.* **45**, 167–256 (2003)
20. Qiu, Y.Q., Zhang, S., Zhang, X.S., Chen, L.: Detecting disease associated modules and prioritizing active genes based on high throughput data. *BMC Bioinform.* **11**, 26 (2010)

21. Wang, J., Zhang, S., Wang, Y., Chen, L., Zhang, X.S.: Disease-aging network reveals significant roles of aging genes in connecting genetic diseases. *PLoS Comput. Biol.* **5**(9), e1000521 (2009)
22. Wang, B., Mezlini, A.M., Demir, F., Fiume, M., Tu, Z., Brudno, M., Haibe-Kains, B., Goldenberg, A.: Similarity network fusion for aggregating data types on a genomic scale. *Nat. Methods* **11**, 333–337 (2014)
23. Cancer Genome Atlas Research Network, Weinstein, J.N., Collisson, E.A., Mills, G.B., Shaw, K.R., Ozenberger, B.A., Ellrott, K., Shmulevich, I., Sander, C., Stuart, J.M.: The Cancer Genome Atlas Pan-Cancer analysis project. *Nat. Genet.* **45**(10), 1113–1120 (2013)
24. Yan, K.K., Wang, D., Rozowsky, J., Zheng, H., Cheng, C., Gerstein, M.: OrthoClust: an orthology-based network framework for clustering data across multiple species. *Genome Biol.* **15**(8), R100 (2014)
25. Zhang, S., Wang, R.S., Zhang, X.S.: Identification of overlapping community structure in complex networks using fuzzy c-means clustering. *Physica A* **374**(1), 483–490 (2007)
26. Zhang, S., Wang, R.S., Zhang, X.S.: Uncovering fuzzy community structure in complex networks. *Phy. Rev. E* **76**, 046103 (2007)
27. Zhang, S., Jin, G., Zhang, X.S., Chen, L.: Discovering functions and revealing mechanisms at molecular level from biological networks. *Proteomics* **7**(16), 2856–2869 (2007)
28. Zhang, S., Ning, X.M., Ding, C., Zhang, X.S.: Determining modular organization of protein interaction networks by maximizing modularity density. *BMC Syst. Biol.* **4**(S2), S10 (2010)
29. Zhang, S., Li, Q., Liu, J., Zhou, X.J.: A novel computational framework for simultaneous integration of multiple types of genomic data to identify microRNA-gene regulatory modules. *Bioinformatics* **27**, 401–409 (2011)
30. Zhang, S., Liu, C.C., Li, W., Shen, H., Laird, P.W., Zhou, X.J.: Discovery of multi-dimensional modules by integrative analysis of cancer genomic data. *Nucleic Acids Res.* **40**, 9379–9391 (2012)
31. Zhang, S., Zhao, J., Zhang, X.S.: Common community structure in time-varying networks. *Phys. Rev. E* **85**, 056110 (2012)
32. Zhang, J., Wu, L.Y., Zhang, X.S., Zhang, S.: Discovery of co-occurring driver pathways in cancer. *BMC Bioinform.* **15**(1), 271 (2014)
33. Zhao, J., Zhang, S., Wu, L.Y., Zhang, X.S.: Efficient methods for identifying mutated driver pathways in cancer. *Bioinformatics* **28**, 2940–2947 (2012)