Intelligent Systems Reference Library 85

Dariusz Król
Damien Fay
Bogdan Gabryś *Editors*

# Propagation Phenomena in Real World Networks

Springer

# Intelligent Systems Reference Library

Volume 85

*About this Series*

The aim of this series is to publish a Reference Library, including novel advances and developments in all aspects of Intelligent Systems in an easily accessible and well structured form. The series includes reference works, handbooks, compendia, textbooks, well-structured monographs, dictionaries, and encyclopedias. It contains well integrated knowledge and current information in the field of Intelligent Systems. The series covers the theory, applications, and design methods of Intelligent Systems. Virtually all disciplines such as engineering, computer science, avionics, business, e-commerce, environment, healthcare, physics and life science are included.

More information about this series at http://www.springer.com/series/8578

Dariusz Król · Damien Fay
Bogdan Gabryś
Editors

# Propagation Phenomena in Real World Networks

Springer

*Editors*

Dariusz Król
Department of Information Systems
Wrocław University of Technology
Wrocław
Poland

Damien Fay
School of Design, Engineering and Computing
Bournemouth University
Fern Barrow Poole
UK

Bogdan Gabryś
Computational Intelligence Research Group,
Smart Technology Research Centre
Bournemouth University
Poole
UK

ISSN 1868-4394 ISSN 1868-4408 (electronic)
Intelligent Systems Reference Library
ISBN 978-3-319-15915-7 ISBN 978-3-319-15916-4 (eBook)
DOI 10.1007/978-3-319-15916-4

Library of Congress Control Number: 2015934449

Springer Cham Heidelberg New York Dordrecht London

Printed on acid-free paper

Springer International Publishing AG Switzerland is part of Springer Science+Business Media (www.springer.com)

# Preface

Propagation phenomena have become a pervasive and significant feature of real world networks. These interdisciplinary phenomena are influencing science, engineering, finance, business, and ultimately society itself. The development of propagation techniques plays an important role in maintaining existing networks and has allowed, for example, synchronization in electrical power grids, prediction of complex system behavior, resource discovery and monitoring, locating biological invasions and assessing damage, virus propagation control and containment, and decomposition and immunization of social and large-scale infrastructure networks. By studying propagation processes, one can better understand information and knowledge spreading in systems which in turn can lead to some improvements in performance and robustness.

The purpose of this book is to bring into one volume the different types of propagation models and techniques including: epidemic models, models for trust inference, coverage strategies for networks, vehicle flow propagation, bio-inspired routing algorithms, P2P botnet attacks and defenses, fault propagation in gene-cellular networks, malware propagation for mobiles, information propagation in crisis situations, financial contagion in interbank networks, and finally how to maximize the spread of influence in social networks. This volume provides a unique compendium of current and emerging problems of propagation and related subjects. Thus, it is truly a guide designed for interdisciplinary use.

Fourteen chapters provided by established scientists in the area of complex networks have been carefully selected to reflect the diversity, complexity, and the depth and breadth of this multidisciplinary area which encompasses closely intertwined conceptual and empirical issues in real world networks.

More precisely, we start with the critical issues regarding epidemics and their outbursts in network structures. Chapter 1 mainly focused on susceptible-infectious-recovered (SIR) and susceptible-infectious-susceptible (SIS) models as well as the various modeling techniques for studying cascading failures, where the damage spreads through the network. Chapter 2 explores an interesting example of information propagation provided by the shoaling and schooling behaviors of fish. The presented fish algorithm (FA), considered as a swarm optimization, using bottom-up learning of individual perception and the propagation of social information in the population, can usefully spread information amongst the agents. In Chap. 3, an approach named Appleseed, which is based on mechanics taken from neuropsychology, known as spreading activation models, is presented. Several algorithms for inferring trust and distrust that go beyond network structure and demonstrate their accuracy in real social networks are evaluated and discussed. Chapter 4 examines how basic network properties can affect the propagation of ideas, beliefs, and behaviors that shape mental models, showing how these different structures can either promote or hinder the adaptation of mental models, and indicating what strategies may be used for improving them. Using methods from statistical mechanics, in Chap. 5, flooding-based strategies are shown to develop random walk strategies to maximize the coverage in large-scale unstructured networks, which takes into account the resource constraints in the form of consumed bandwidth and latency time.

Modeling propagation in transport and communication networks is the central topic of the next two contributions. The first of these chapters, Chap. 6 utilizes three kinds of Petri nets: place/transitions Petri nets, timed Petri nets, and hybrid Petri nets to model and simulate agents of a transport network. A similar line of research is explored in Chap. 7 but the focus is on the role of propagation phenomena in bio-inspired routing.

Another popular area where propagation methods are highly topical is building reliable systems with respect to errors, faults, and failures. This is the focus of Chap. 8, where the life cycle of P2P botnets is studied. This chapter provides guidance for security professionals on how to implement two mitigation techniques against P2P botnets index poisoning defense and Sybil defense, and one monitoring technique—passive monitoring—to achieve better performance. Network modifications with the aim of enhancing robustness against targeted attacks is proposed in Chap. 9. The outlined procedure optimized for the cost function of Integral Efficiency could be used to generate highly robust and efficient networks. Chapter 10 in turn examines problems related to propagation phenomena in biological networks. As an example, carcinogenesis and cancer progression are examined as processes that propagate molecular failures. These processes are based on evolutionary and spatial evolutionary games, which describe propagation phenomena in time and space. A number of recently developed propagation models and algorithms that could be utilized to understand mobile phone datasets are presented in Chap. 11. It discusses how to analyze the spreading dynamics of mobile phone malware using SIR, SIS, and SIDR epidemiological models, how to identify fraudulent activity over a phone network, and how to predict churners in a phone network. Chapter 12 deals with structural information propagation in crisis situations referring to disaster, emergency, or catastrophe management loosely defined as all actions taken before, during, and after the event. It proposes a framework for detecting and propagating valuable information to the public, and to the first responders in particular. A model of a financial network and the three methods of the propagation of losses: the linear threshold algorithm, the graph-theoretic approach, and the fictitious default algorithm, used to simulate contagion in such a network is proposed in

Chap. 13. The volume closes with Chap. 14. It presents the state of the art in the area of maximizing the spread of influence in social networks, limitations of using a static network representation, and recent trends that use temporal properties of social networks as an alternative.

We are extremely grateful to the Intelligent Systems Reference Library by Springer for having hosted this theme book. Special thanks are due to series editors: Janusz Kacprzyk, and Lakhmi C. Jain. We would principally like to express our most sincere thanks and great appreciation to all those colleagues who have helped us in the realization of this book, in particular, to the contributors and referees.

Many researchers have contributed to this volume with their work. We deeply thank them for their great contributions. In alphabetical order they are: Baber Aslam, Damian Borys, Abdelhamid Bouchachia, Anthony Brabazon, Newton Paulo Bueno, František Čapkovič, Wei Cui, Fabio Daolio, Derek Doran, Mario Eboli, Niloy Ganguly, Jennifer Golbeck, Hans J. Herrmann, Roman Jaksik, Przemysław Kazienko, Pavel Krömer, Michał Krześlak, Vitor H.P. Louzada, Veena Mendiratta, Radosław Michalski, Petr Musilek, Subrata Nandi, Michael ONeill, Daniela Pohl, Jarosław Śmieja, Andrzej Świerniak, Somnath Tagore, Marco Tomassini, Ping Wang, Lei Wu, Cai-Nicolas Ziegler, and Cliff C. Zou.

Warm thanks are also due to the following referees who reviewed the chapters with remarkable expertise and engagement: Nuno Araujo, Francesca Arcelli Fontana, Emili Balaguer-Ballester, František Čapkovič, Richard Clegg, Ireneusz Czarnowski, Anirban Dasgupta, Paul Davidsson, Damien Fay, Evelina Gabasova, Bogdan Gabryś, Sergio Gomez, Hamed Haddadi, Jason Jung, Dariusz Król, Hui Li, Katarzyna Musial, Marco Rossetti, Ruben Sanchez-Garcia, Antonio Scala, Fabio Stella, Mirko Viroli, Ye Wu, and Rong Yang.

By way of conclusion, it is worth emphasizing that this work was supported by the project *To what extent can design principles for complex networks be derived from the study of error propagation phenomenon in smart and bio-inspired network structures?* within the 7th European Community Framework Programme under grant FP7-PEOPLE-2010-IEF-274375-EPP. Indeed, the true situation is barely captured by saying that without the EC project it would not have been possible to start, and no less complete, this volume.

We anticipate that our work results in a coherent and comprehensive presentation of the vast recent research activity concerning propagation processes in real world networks. The large number of citations that found room in every chapter makes us believe that the present volume will be a convenient reference to all scholars who consider studying this exciting research area. It should be also clear that these 14 chapters should not be construed as covering all aspects of propagation research.

Wrocław, November 2014
Bournemouth

Dariusz Król
Damien Fay
Bogdan Gabryś

# Contents

# Contributors

**Baber Aslam** National University of Sciences and Technology, Islamabad, Pakistan

**Damian Borys** Systems Engineering Group, Faculty of Automatic Control, Electronics and Informatics, Silesian University of Technology, Gliwice, Poland

**Abdelhamid Bouchachia** Smart Technology Research Center, Bournemouth University, Fern Barrow Poole, UK

**A. Brabazon** Complex Adaptive Systems Laboratory and School of Business, University College Dublin, Dublin, Ireland

**Newton Paulo Bueno** UFV-Federal University of Vicosa, Vicosa, Minas Gerais, Brazil

**František Čapkovič** Institute of Informatics, Slovak Academy of Sciences, Bratislava, Slovakia

**W. Cui** Complex Adaptive Systems Laboratory and School of Business, University College Dublin, Dublin, Ireland

**Fabio Daolio** Faculty of Business and Economics, University of Lausanne, Lausanne, Switzerland

**Derek Doran** Department of Computer Science and Engineering, Kno.e.sis Research Center, Wright State University, Dayton, OH, USA

**Mario Eboli** Dipartimento di Economia Aziendale, Universitá 'G. d'Annunzio', Pescara, Italy

**Niloy Ganguly** Indian Institute of Technology, Kharagpur, India

**Jennifer Golbeck** University of Maryland, College Park, MD, USA

**Hans J. Herrmann** Computational Physics, IfB, ETH Zurich, Zurich, Switzerland; Departamento de Física, Universidade Federal Do Ceará, Fortaleza, Ceará, Brazil

**Roman Jaksik** Systems Engineering Group, Faculty of Automatic Control, Electronics and Informatics, Silesian University of Technology, Gliwice, Poland

**Przemysław Kazienko** Department of Computational Intelligence, Wrocław University of Technology, Wrocław, Poland

**Pavel Krömer** Faculty of Electrical Engineering and Computer Science, VŠB Technical University of Ostrava, Ostrava, Czech Republic

**Michał Krześlak** Systems Engineering Group, Faculty of Automatic Control, Electronics and Informatics, Silesian University of Technology, Gliwice, Poland

**Vitor H.P. Louzada** Computational Physics, IfB, ETH Zurich, Zurich, Switzerland

**Veena Mendiratta** Bell Labs, Alcatel-Lucent, Naperville, IL, USA

**Radosław Michalski** Department of Computational Intelligence, Wrocław University of Technology, Wrocław, Poland

**Petr Musilek** Department of Electrical and Computer Engineering, University of Alberta, Edmonton, Canada

**Subrata Nandi** National Institute of Technology, Durgapur, India

**M. O'Neill** Complex Adaptive Systems Laboratory and School of Business, University College Dublin, Dublin, Ireland

**Daniela Pohl** Institute of Information Technology, Alpen-Adria-Universität Klagenfurt, Klagenfurt, Austria

**Jarosław Śmieja** Systems Engineering Group, Faculty of Automatic Control, Electronics and Informatics, Silesian University of Technology, Gliwice, Poland

**Andrzej Świerniak** Systems Engineering Group, Faculty of Automatic Control, Electronics and Informatics, Silesian University of Technology, Gliwice, Poland

**Somnath Tagore** Department of Biotechnology and Bioinformatics, Padmashree Dr. D.Y. Patil University, Navi Mumbai, India

**Marco Tomassini** Faculty of Business and Economics, University of Lausanne, Lausanne, Switzerland

**Ping Wang** Symantec Corporation, Florida, USA

**Lei Wu** Department of Computer Science, North Carolina State University, Raleigh, NC, USA

**Cai-Nicolas Ziegler** XING EVENTS GmbH, Munich, Germany

**Cliff C. Zou** Department of Electrical Engineering and Computer Science, University of Central Florida, Orlando, USA

# Chapter 1
# Epidemic Models: Their Spread, Analysis and Invasions in Scale-Free Networks

**Somnath Tagore**

**Abstract** The mission of this chapter is to introduce the concept of epidemic outbursts in network structures, especially in case of scale-free networks. The invasion phenomena of epidemics have been of tremendous interest among the scientific community over many years, due to its large scale implementation in real world networks. This chapter seeks to make readers understand the critical issues involved in epidemics such as propagation, spread and their combat which can be further used to design synthetic and robust network architectures. The primary concern in this chapter focuses on the concept of Susceptible-Infectious-Recovered (SIR) and Susceptible-Infectious-Susceptible (SIS) models with their implementation in scale-free networks, followed by developing strategies for identifying the damage caused in the network. The relevance of this chapter can be understood when methods discussed in this chapter could be related to contemporary networks for improving their performance in terms of robustness. The patterns by which epidemics spread through groups are determined by the properties of the pathogen carrying it, length of its infectious period, its severity as well as by network structures within the population. Thus, accurately modeling the underlying network is crucial to understand the spread as well as prevention of an epidemic. Moreover, implementing immunization strategies helps control and terminate theses epidemics.

## 1.1 Scale-Free Networks

The degree distribution of individuals is one of the most standard and efficient network measures that is existent today. In most of the synthetic as well as practical networks, many individuals have lesser number of connected neighbours than others.

S. Tagore (✉)
Department of Biotechnology and Bioinformatics,
Padmashree Dr. D.Y. Patil University, Navi Mumbai 400614, India
e-mail: somnathtagore@yahoo.co.in

D. Król et al. (eds.), *Propagation Phenomena in Real World Networks*,
Intelligent Systems Reference Library 85, DOI 10.1007/978-3-319-15916-4_1

For instance, random networks, small worlds display lesser variation in terms of neighbourhood sizes, whereas spatial networks have Poisson-like degree distributions. Moreover, as highly connected individuals are of more importance considering disease transmission, incorporating them into the current network is of outmost importance [4]. This is essential in case of capturing the complexities of disease spread. Architecturally, scale-free networks are heterogenous in nature and can be dynamically constructed by adding new individuals to the current network structure one at a time. This strategy is similar to naturally forming links, especially in case of social networks. Moreover, the newly connected nodes or individuals link to the already existent ones (with larger connections) in a manner that is preferential in nature. This connectivity can be understood by a power-law plot with the number of contacts per individual, a property which is regularly observed in case of several other networks like that of power grids, world-wide-web, to name a few [14].

Epidemiologists have worked hard on understanding the heterogeneity of scale-free networks for populations for a long time. Highly connected individuals as well as hub participants have played essential roles in the spread and maintenance of infections and diseases. Figure 1.1 illustrates the architecture of a system consisting of a population of individuals. It has several essential components, namely, nodes, links, newly connected nodes, hubs and sub-groups respectively. Here, nodes correspond to individuals and their relations are shown as links. Similarly, newly connected nodes correspond to those which are recently added to the network, such as initiation of new relations between already existing and unknown individuals [24]. Hubs are

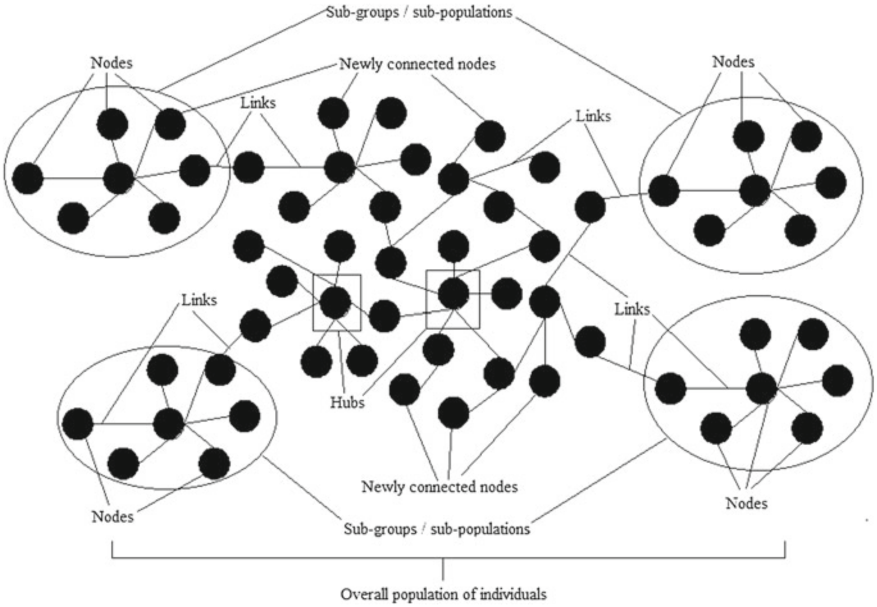

Sub-groups / sub-populations
Nodes
Newly connected nodes
Nodes
Links
Links
Links
Links
Hubs
Newly connected nodes
Nodes
Nodes
Sub-groups / sub-populations
Overall population of individuals

**Fig. 1.1** A synthetic scale-free network and its characteristics

those nodes which are highly connected, such as individuals who are very popular among others and have many relations and/or friends. Lastly, sub-groups correspond to certain sections of the population which have individuals with closely associated relationships, such as group of nodes which are highly dense in nature, or having high clustering coefficient. Furthermore, it is important in having large number of contacts as the individuals are at greater risk of infection and, once infected, can transmit it to others. For instance, hub individuals of such high-risk individuals help in maintaining sexually transmitted diseases (STDs) in different populations where majority belong to long-term monogamous relationships, whereas in case of SARS epidemic, a significant proportion of all infections are due to high risk connected individuals. Furthermore, the preferential attachment model proposed by Barabási and Albert [4] defined the existence of individuals of having large connectivity does not require random vaccination for preventing epidemics. Moreover, if there is an upper limit on the connectivity of individuals, random immunization can be performed to control infection.

Likewise, the dynamics of infectious diseases has been extensively studied in case of scale-free as well as small-world and random networks. In small-world networks, most of the nodes may not be direct neighbors, but can be reached from all other nodes via less number of hops, that are number of nodes between start and terminating nodes. Also, in these networks distance, *dist*, between two random nodes increases proportionally to the logarithm of the number of nodes, *tot*, in the network [15], i.e.,

$$dist \propto \log tot \tag{1.1}$$

Watts and Strogatz [24] identified a class of small-world networks and categorized them as random graphs. These were classified on the basis of two independent features, namely, average shortest path length and clustering coefficient. As per Erdős-Rényi model, random graphs have a smaller average shortest path length and small clustering coefficient. Watts and Strogatz on the other hand demonstrated that various real-world networks have a smaller average shortest path length along with high clustering coefficient greater than expected randomly. It has been observed that it is difficult to block and/or terminate an epidemic in scale-free networks with slow tails. It has especially been seen in case the network correlations among infections and individuals are absent. Another reason for this effect is the presence of hubs, where infections could be sustained and reduced by target-specific selections [17].

### 1.1.1 Power-Law

It has been well known that real-world networks ranging from social to computers are scale-free in nature, whose degree distribution follows an asymptotic power-law. These are characterized by degree distribution following a power law,

$$P(conn) \approx conn^{-\eta} \tag{1.2}$$

for the number of connections, *conn* for individuals and η is an exponent. Barabási and Albert [4] analyzed the topology of a portion of the world-wide-web and identified 'hubs'. The terminals had larger number of connections than others and the whole network followed a power-law distribution. They also found that these networks have heavy-tailed degree distributions and thus termed them as 'scale-free'. Likewise, models for epidemic spread in static heavy-tailed networks have illustrated that with a degree distribution having moments resulted in lesser prevalence and/or termination for smaller rates of infection [14]. Moreover, beyond a particular threshold, this prevalence turns to non-zero. Similarly, it has been seen that for networks following power-law,

$$\mathit{moment} > \eta - 1 \tag{1.3}$$

does not exist and the prevalence is non-zero for any infection rates. Due to this reason, epidemics are difficult to handle and terminate in static networks having power-law degree distributions. Figure 1.2 illustrates a power-law plot between $P(conn)$

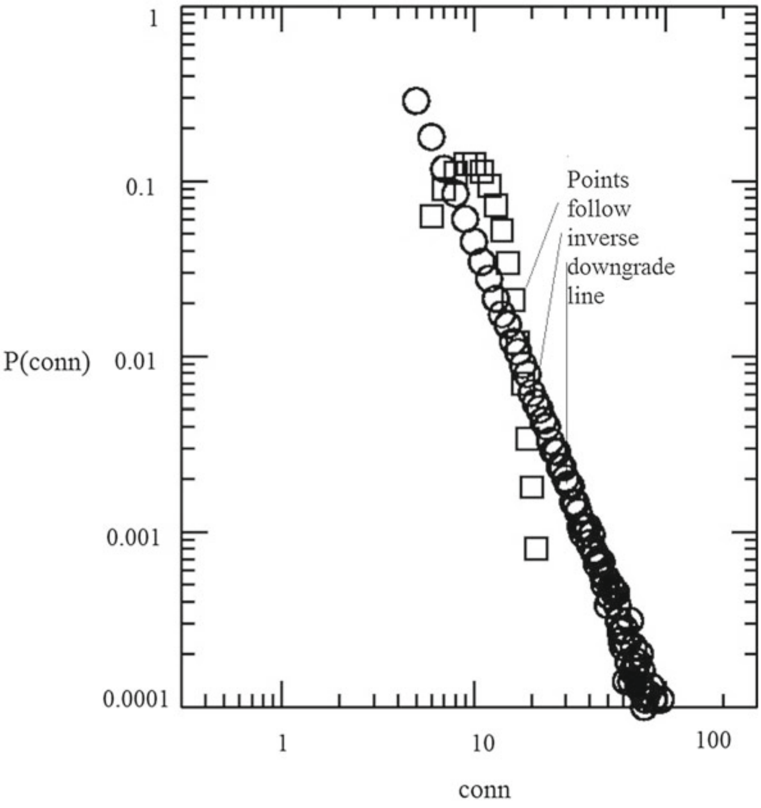


**Fig. 1.2** Power-law curve illustrating $P(conn)$ versus *conn* in log-log scale

versus *conn* in log-log scale. It shows that points in this figure follow a inverse downgrade line in log-log scale, satisfying ‘scale-free’ behavior.

Likewise, in various instances, networks are not static but dynamic (i.e., they evolve in time) via some rewiring processes, in which edges are detached and reattached according to some dynamic rule. Steady states of rewiring networks have been studied in the past. More often, it has been observed that depending on the average connectivity and rewiring rates, networks reach a scale-free steady state, with an exponent, $\eta$, represented using dynamical rates [17].

## 1.2 Epidemics

The study of epidemics has always been of interest in areas where biological applications coincide with social issues. For instance, epidemics like influenza, measles, and STDs, can pass through large group of individuals, populations, and/or persist over longer timescales at low levels. These might even experience sudden changes of increasing and decreasing prevalence. Furthermore, in some cases, single infection outbreaks may have significant effects on a complete population group [1].

Epidemic spreading can also occur on complex networks with vertices representing individuals and the links representing interactions among individuals. Thus, spreading of diseases can occur over the network of individuals as spreading of computer viruses occur over the world-wide-web. The underlying network in epidemic models is considered to be static while the individual states vary from infected to non-infected individuals according to certain probabilistic rules. Furthermore, the evolution of an infected group of individuals in time can be studied by focusing on the average density of infected individuals in steady state. Lastly, the spread as well as growth of epidemics can also be monitored by studying the architecture of the network of individuals as well as its statistical properties [2].

### *1.2.1 Branching*

One of the essential properties of epidemic spread is its branching pattern, thereby infecting healthy individuals over a time period. This branching pattern of epidemic progression can be classified on the basis of their infection initiation, spread and further spread (Fig. 1.3) [5].

1. Infection initiation: If an infected individual comes in contact with a group of individuals, the infection is transmitted to each with a probability $p$, independent of one another. Furthermore, if the same individual meets $k$ others while being infected, these $k$ individuals form the infected set. Due to this random disease transmission from the initially infected individual, those directly connected to it get infected.

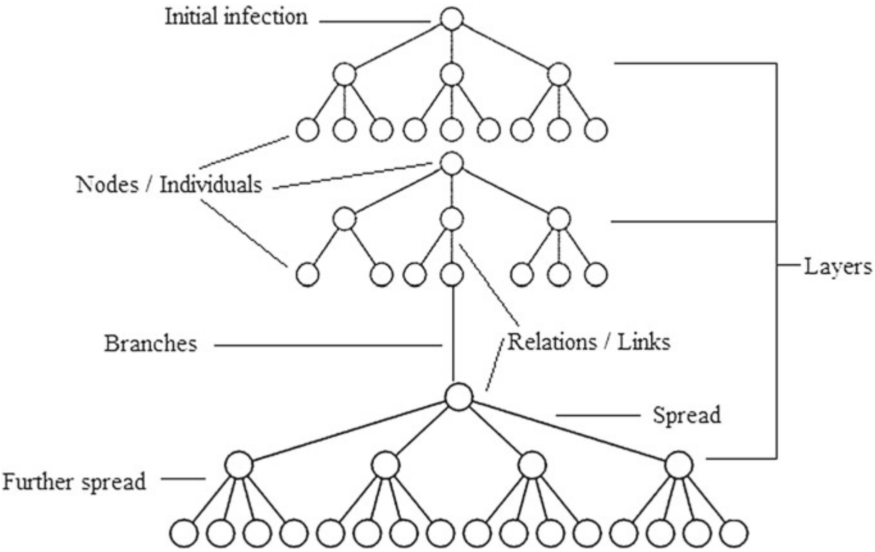


**Fig. 1.3** Branching modes and patterns in epidemic progression

2. Spread: Every individual in the original infected set meets $k$ other individuals, which results in $k^2$ individuals.
3. Further spread: The infection spreads further with each individual in the present infected set connecting to $k$ healthy individuals with a probability $p$ independent of individual infection.

#### 1.2.1.1 Reproductive Number

If infection in a branching process reaches an individual set and fails to infect healthy individuals, then termination of the infection occurs, which leads to no further progression and infection of other healthy individuals. Thus, there may be two possibilities for an infection in a branching process model. Either it reaches a site infecting no further and terminating out, or it continues to infect healthy individuals through contact processes. The quantity which can be used to identify whether an infection persist or fades out is defined as *basic reproductive number* [6].

This basic reproductive number, τ, is the expected number of newly infected individuals caused by a single already infected individual. In case where every individual meets $k$ new people and infects each with probability $p$, the basic reproductive number is represented as

$$\tau = pk \tag{1.4}$$

It is quite essential as it helps in identifying whether or not an infection can spread through a population of healthy individuals. The concept of τ was first proposed by Alfred Lotka, and applied in the area of epidemiology by MacDonald [13].

For non-complex population models, τ can be identified if information for 'death rate' is present. Thus, considering death rate, $d$, and birth rate, $b$, at the same time,

$$\tau = \frac{b}{d} \tag{1.5}$$

Moreover, τ can also be used to determine whether an infection will terminate, i.e., $\tau < 1$ or it becomes an epidemic, i.e., $\tau > 1$. But, it cannot be used for comparing different infections at the same time on the basis of multiple parameters. Several methods, such as identifying eigenvalues, Jacobian matrix, birth rate, equilibrium states, population statistics can well be used to analyze and handle τ [18].

#### 1.2.1.2 Branching Models

There are some standard branching models that are existent for analyzing the progress of infection in a healthy population or network. The first one, *Reed-Frost model*, considers a homogeneous close set consisting of total number of individuals, *tot*. Let *num* designate the number of individuals susceptible to infection at time $t = 0$ and $m_{num}$ the number of individuals infected by the infection at any time $t$ [19]. Here,

$$num + m_{num} = tot \tag{1.6}$$

$$m_{num} = num \tag{1.7}$$

Here, Eq. 1.7 is in case of a smaller population. It is assumed that an individual $x$ is infected at time $t$, whereas any individual $y$ comes in contact with $x$ with a probability $\frac{a}{num}$, where $a > 0$. Likewise, if $y$ is susceptible to infection then it becomes infected at time $t + 1$ and $x$ is removed from the population (Fig. 1.4a). In this figure, $x$ or $v_1(*)$ represents the infection start site, $y(v_3)$, $v_2$ are individuals that are susceptible to infection, $num = 0$, $tot = 11$, and $m_{num} = 1$.

The second one, *3-clique model* constructs a 3-clique sub-network randomly by assigning a set of *tot* individuals. Here, for individual/vertex pair $(v_i, v_j)$ with probability $p_1$, the pair is included along with vertices triples $(v_i, v_j, v_k)$ with probability $p_2$. Thus, the corresponding pairs $(v_i, v_j)$, $(v_j, v_k)$ and $(v_k, v_i)$ are also included. This creates a network

$$G = g_1 \bigcup g_2 \tag{1.8}$$

Here, $g_1, g_2$ are two independent graphs, where $g_1$ is a Bernoulli graph with edge probability $p_1$ and $g_2$ with all possible triangles existing independently with a probability $p_2$ (Fig. 1.4b). In this figure, $g_1 = (v_1, v_2, v_3)$, $g_2 = (v_4, v_5, v_6)$, $g_3 = (v_7, v_8, v_9)$ are the three 3-clique sub-networks with $tot = 9$, and $G = g_1 \bigcup g_2 \bigcup g_3$ respectively [21].

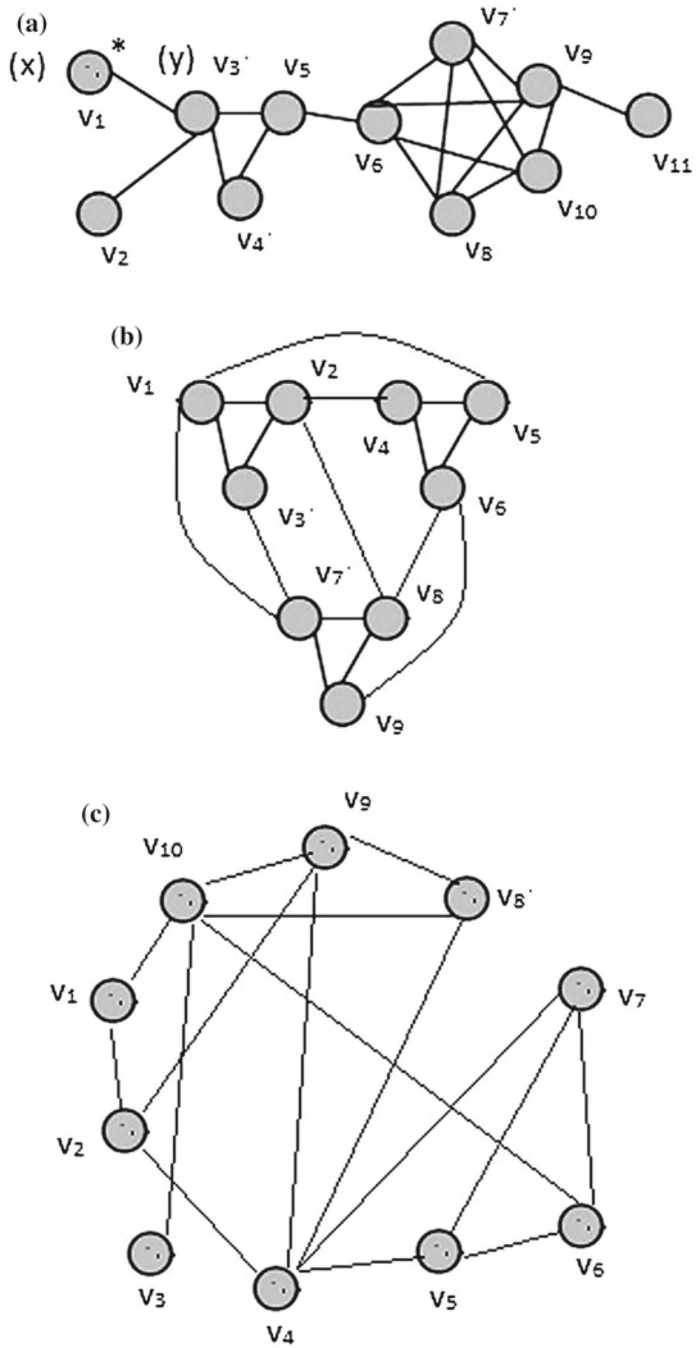


**Fig. 1.4** Types of branching models illustrated in synthetic networks: **a** *Reed-Frost*, **b** *3-clique*, **c** *Household*

The third one, *Household model* assumes that for a given a set of *tot* individuals or vertices, $g_1$ is a Bernoulli graph consisting of $\frac{tot}{b}$ disjoint $b-$cliques, where $b \ll tot$ with edge probability $p_2$. Thus, the network $G$ is formed as the superposition of the graphs $g_1$ and $g_2$, i.e., $G = g_1 \bigcup g_2$. Moreover, $g_1$ fragments the population into mutually exclusive groups whereas $g_2$ describes the relations among individuals in the population. Thus, $g_1$ does not allow any infection spread, as there are no connections between the groups. But, when the relationship structure $g_2$ is added, the groups are linked together and the infection can now spread using relationship connections (Fig. 1.4c). In this figure, $tot = 10$ where the individuals ($v_1$ to $v_{10}$) are linked on the basis of randomly assigned $p_2$ and $b = 4 \ll tot = 10$.

## 1.3 Network Architectures

The interconnected architecture of various networks have been of primary interest to researchers in various scientific areas. In interconnected networks, failure in vertex links in one network can cause failure of dependent vertices in other networks. This results in cascading failures. Similarly, in case of networks without dependencies among vertices, the level of information flow between the interconnected vertices affects the epidemic transition on subset levels. Furthermore, percolation threshold in interacting networks are lower than in single networks, with the appearance of a giant component in certain cases (Fig. 1.5). A giant component is a connected sub-graph of a random graph containing a constant fraction of total vertices of the entire graph. These are extremely prominent in Erdős-Rényi graphs, where each edge connecting vertex pairs for a set of $n$ vertices remains independently of one another with a probability $p$. Here, if $p \leq \frac{1-\varepsilon}{n}$ for any constant $\varepsilon > 0$, then all the connected components have size $O(\log n)$, and giant component is absent. But, for $p \geq \frac{1+\varepsilon}{n}$ a single giant component may reside. Figure 1.5a–d illustrate the formation of a giant component in a random graph with $p = 0.002, 0.006, 0.009$ in Fig. 1.5b–d respectively [23].

Thus, it is essential to identify the conditions which results in an epidemic spread in one network, with the presence of minimal isolated infections on other network components. Moreover, depending on the parameters of individual sub-networks and their internal connectivities, connecting them to one another creates marginal effect on the spread of epidemic. Thus, identifying these conditions resulting in analyzing spread of epidemic process is very essential. In this case, two different interconnected network modules can be determined, namely, strongly and weakly coupled. In the strongly coupled one, all modules are simultaneously either infection free or part of an epidemic, whereas in the weakly coupled one a new mixed phase exists, where the infection is epidemic on only one module, and not in others [25].

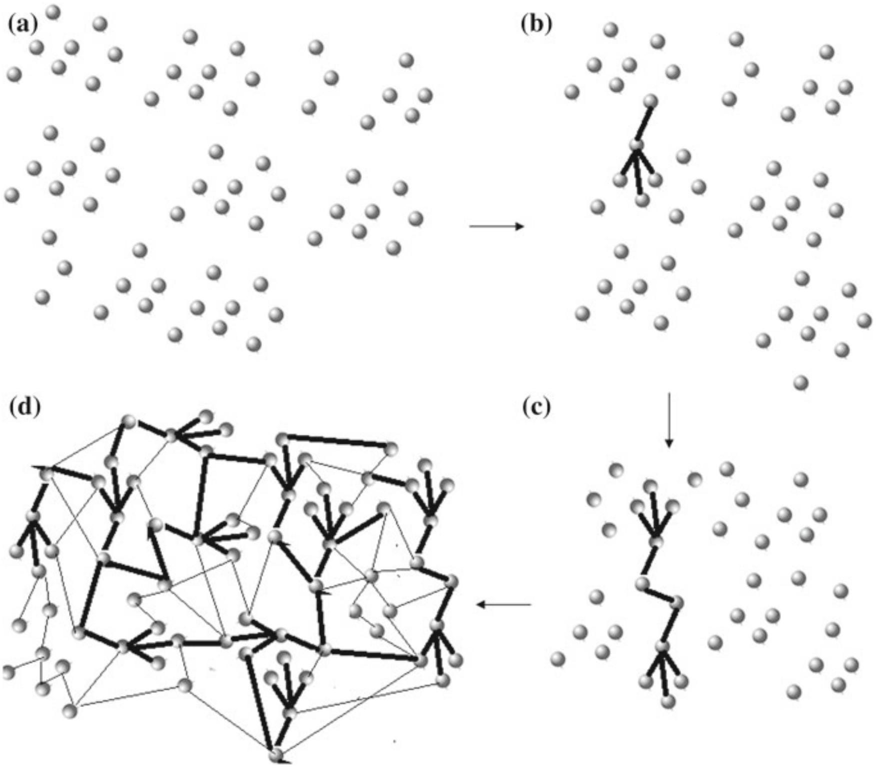


**Fig. 1.5** Emergence of giant component in an interconnected network. **a** Original network, **b** emergence of giant component, **c** further emergence, and **d** final architecture

### *1.3.1 Concurrency*

Generally, epidemic models consider contact networks to be static in nature, where all links are existent throughout the infection course. Moreover, a property of infection is that these are contagious and spread at a rate faster than the initially infected contact. But, in cases like HIV, which spreads through a population over longer time scales, the course of infection spread is heavily dependent on the properties of the contact individuals. The reason for this being, certain individuals may have lesser contacts at any single point in time and their identities can shift significantly with the infection progress [25].

Thus, for modeling the contact network in such infections, transient contacts are considered which may not last through the whole epidemic course, but only for particular amount of time. In such cases, it is assumed that the contact links are undirected. Furthermore, different individual timings do not affect those having potential to spread an infection but the timing pattern also influences the severity of the overall epidemic spread. Similarly, individuals may also be involved in concurrent partnerships having two or more actively involved ones that overlap in time. Thus, the concurrent pattern causes the infection to circulate vigorously through the network [22].

## 1.4 Propagation Phenomena in Real World Networks

In the last decade, considerable amount of work has been done in characterizing as well as analyzing and understanding the topological properties of networks. It has been established that scale-free behavior is one of the most fundamental concepts for understanding the organization various real-world networks. This scale-free property has a resounding effect on all aspect of dynamic processes in the network, which includes percolation. Likewise, for a wide range of scale-free networks, epidemic threshold is not existent, and infections with low spreading rate prevail over the entire population [10]. Furthermore, properties of networks such as topological fractality etc. correlate to many aspects of the network structure and function. Also, some of the recent developments have shown that the correlation between degree and betweenness centrality of individuals is extremely weak in fractal network models in comparison with non-fractal models [20].

Likewise, it is seen that fractal scale-free networks are dis-assortative, making such scale-free networks more robust against targeted perturbations on hubs nodes. Moreover, one can also relate fractality to infection dynamics in case of specifically designed deterministic networks. Deterministic networks allow computing functional, structural as well as topological properties. Similarly, in case of complex networks, determination of topological characteristics has shown that these are scale-free as well as highly clustered, but do not display small-world features. Also, by mapping a standard Susceptible, Infected, Recovered (SIR) model to a percolation problem, one can also find that there exists certain finite epidemic threshold. In certain cases, the transmission rate needs to exceed a critical value for the infection to spread and prevail. This also specifies that the fractal networks are robust to infections [11]. Meanwhile, scale-free networks exhibit various essential characteristics such as power-law degree distribution, large clustering coefficient, large-world phenomenon, to name a few [16].

## 1.5 Network Definition and Measurement

Network analysis can be used to describe the evolution and spread of information in the populations along with understanding their internal dynamics and architecture. Specifically, importance should be given to the nature of connections, and whether a relationship between *x* and *y* individuals provide a relationship between *y* and *x* as well. Likewise, this information could be further utilized for identifying transitivity-based measures of cohesion (Fig. 1.6).

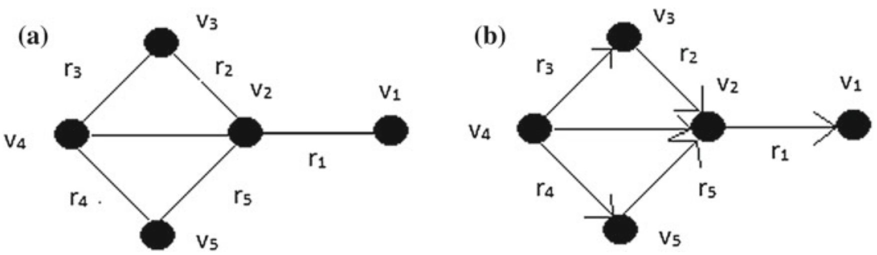


**Fig. 1.6** Architectural properties in a hypothetical. **a** Undirected, **b** directed network

Meanwhile, research in networks also provide some quantitative tools for describing and characterizing networks. *Degree* of a vertex is the number of connectivities for each vertex in the form of links. For instance, $degree(v_4) = 3$, $degree(v_2) = 4$ (for undirected graph (Fig. 1.6a)). Similarly for Fig. 1.6b, $degree_{in}(v_2) = 3$ (number of incoming links), $degree_{out}(v_2) = 1$ (number of outgoing links). *Clustering coefficient (CC)* of a vertex is the compactness of the network, i.e., $CC(v_i) = \frac{2*link}{degree(degree-1)}$, where *degree* = degree of vertex $v_i$, *link* = number of links among neighbors of $v_i$. For instance, in Fig. 1.6a, $CC(v_2) = 0.33$, $CC(v_4) = 0.6$, etc. Likewise, *Shortest path* is the minimum number of links that needs to be parsed for traveling between two vertices. For instance, in Fig. 1.6a, shortest path between $v_4$ and $v_1 = (v_4, v_2, v_1)$. *Diameter* of network is the maximum distance between any two vertices or the longest of the shortest walks. Thus, in Fig. 1.6b, from $v_4$, one has $(v_4, v_3, v_2, v_1)$, $(v_4, v_2, v_1)$, $(v_4, v_5, v_2, v_1)$, from $v_3$, we have $(v_3, v_4, v_5, v_2, v_1)$, $(v_3, v_4, v_2, v_1)$, $(v_3, v_2, v_1)$, from $v_5$, we have $(v_5, v_4, v_3, v_2, v_1)$, $(v_5, v_4, v_2, v_1)$, $(v_5, v_2, v_1)$. Out of these the longest of the shortest walks = $(v_3, v_4, v_5, v_2, v_1)$, $(v_5, v_4, v_3, v_2, v_1) = 4$. Thus, diameter = 4 [15].

*Radius* of a network is the minimum eccentricity (eccentricity of a vertex $v_i$ is the greatest geodesic distance), i.e., distance between two vertices in a network is the number of edges in a shortest path connecting them between $v_i$ and any other vertex of any vertex. For instance, in Fig. 1.6b, radius of network = 2. *Betweenness centrality* $(g(v_i))$ is equal to the number of shortest paths from all vertices to all others that pass through vertex $v_i$, i.e.,

$$g(v_i) = \frac{v_x v_y (v_i)}{v_x v_y} \tag{1.9}$$

where $v_x v_y$ is total number of shortest paths from vertex $v_x$ to vertex $v_y$ and $v_x v_y(v_i)$ is the number of those paths that pass through $v_i$. Thus, in Fig. 1.6b, $g(v_4) = 0.77$. Similarly, *Closeness centrality* $(c(v_i))$ of a vertex $v_i$ describes the total distance of $v_i$ to all other vertices in the network, i.e., sum the shortest paths of $v_i$ to all other vertices in the network. For instance, in Fig. 1.6b, $c(v_4) = (v_4, v_3, v_2, v_1) + (v_4, v_2, v_1) + (v_4, v_5, v_2, v_1) = 8$. Lastly, *Stress centrality* $(s(v_i))$ is the simple accumulation of the number of shortest paths between all vertex pairs, sometimes interchangeable with betweenness centrality [14].

Use of ‘adjacency matrix’, $A_{v_i v_j}$, describing the connections within a population is also persistent. Likewise, various network quantities can be ascertained from the adjacency matrix. For instance, size of a population is defined as the average number of contacts per individual, i.e.,

$$\overline{num} = \frac{1}{tot} \sum_{v_i v_j} A_{v_i v_j} \tag{1.10}$$

The powers of adjacency matrix can be used to calculate measures of transitivity [14].

## *1.5.1 Data Collection Process*

One of the key pre-requisites of network analysis is initial data collection. For performing a complete mixing network analysis for individuals residing in a population, every relationship information is essential. This data provides great difficulty in handling the entire population, as well as handling complicated network evaluation issues. The reason being, individuals have contacts, and recall problems are quite probable. Moreover, evaluation of contacts requires certain information which may not always be readily present. Likewise, in case of epidemiological networks, connections are included if they explain relationships capable of permitting the transfer of infection. But, in most of the cases, clarity of defining such relations is absent. Thus, various types of relationships bestow risks and judgments that needs to be sorted for understanding likely transmission routes. One can also consider weighted networks in which links are not merely present or absent but are given scores or weights according to their strength [9].

Furthermore, different infections are passed by different routes, and a mixing network is infection specific. For instance, a network used in HIV transmission is different from the one used to examine influenza. Similarly, in case of airborne infections like influenza and measles, various networks need to be considered because differing levels of interaction are required to constitute a contact. The problems with network definition and measurement imply that any mixing networks that are obtained will depend on the assumptions and protocols of the data collection process.

Three main standard techniques can be employed to gather such information, namely, *infection searching*, *complete contact searching* and *diary-based studies* [9].

### 1.5.1.1 Infection Searching

After an epidemic spread, major emphasis is laid on determining the source and spread of infection. Thus, each infected individual is linked to one other from whom infection is spread as well as from whom the infection is transmitted. As all connections represent actual transmission events, infection searching methods do not suffer from problems with the link definition, but interactions not responsible for this infection transmission are removed. Thus, the networks observed are of closed architecture, without any loops, walks, cliques and complete sub-graphs [15].

Infection searching is a preliminary method for infectious diseases with low prevalence. These can also be simulated using several mathematical techniques based on differential equations, control theories etc., assuming a homogeneous mixing of population. It can also be simulated in a manner so that infected individuals are identified and cured at a rate proportional to the number of neighbors it has, analogous to the infection process. But, it does not allow to compare various infection searching budgets and thus a discrete-event simulation need to be undertaken. Moreover, a number of studies have shown that analyses based on realistic models of disease transmission in healthy networks yields significant projections of infection spread than projections created using compartmental models [8]. Furthermore, depending on the number of contacts for any infected individuals, their susceptible neighbors are traced and removed. This is followed by identifying infection searching techniques that yields different numbers of newly infected individuals on the spread of the disease.

#### 1.5.1.2 Complete Contact Searching

Contact searching identifies potential transmission contacts from an initially infected individual by revealing some new individual set who are prone to infection and can be subject of further searching effort. Nevertheless, it suffers from network definition issues; is time consuming and depends on complete information about individuals and their relationships. It has been used as a control strategy, in case of STDs. Its main objective of contact searching is identifying asymptomatically infected individuals who are either treated or quarantined.

Complete contact searching deals with identifying the susceptible and/or infected individuals of already infected ones and conducting simulations and/or testing them for degree of infection spread, treating them as well as searching their neighbors for immunization. For instance, STDs have been found to be difficult for immunization. The reason being, these have specifically long asymptomatic periods, during which the virus can replicate and the infection is transmitted to healthy, closely related neighbors. This is rapidly followed by severe effects, ultimately leading to the termination of the affected individual. Likewise, recognizing these infections as global epidemic has led to the development of treatments that allow them to be managed by suppressing the replication of the infection for as long as possible. Thus, complete contact searching act as an essential strategy even in case when the infection seems incurable [7].

#### 1.5.1.3 Diary-Based Studies

Diary-based studies consider individuals recording contacts as they occur and allow a larger number of individuals to be sampled in detail. Thus, this variation from the population approach of other tracing methods to the individual-level scale is possible. But, this approach suffers from several disadvantages. For instance, the data collection is at the discretion of the subjects and is difficult for researchers to link this information into a comprehensive network, as the individual identifies contacts that are not uniquely recorded [3].

Diary-based studies require the individuals to be part of some coherent group, residing in small communities. Also, it is quite probable that this kind of a study may result in a large number of disconnected sub-groups, with each of them representing some locally connected set of individuals. Diary-based studies can be beneficial in case of identifying infected and susceptible individuals as well as the degree of infectivity. These also provide a comprehensive network for diseases that spread by point-to-point contact and can be used to investigate the patterns infection spread.

## 1.6 Robustness

Robustness is an essential connectivity property of power-law graph. It defines that power-law graphs are robust under random attack, but vulnerable under targeted attack. Recent studies have shown that the robustness of power-law graph under random and targeted attacks are simulated displaying that power-law graphs are very robust under random errors but vulnerable when a small fraction of high degree vertices or links are removed. Furthermore, some studies have also shown that if vertices are deleted at random, then as long as any positive proportion remains, the graph induced on the remaining vertices has a component of order of the total number of vertices [15].

Many a times it can be observed that a network of individuals may be subject to sudden change in the internal and/or external environment, due to some perturbation events. For this reason, a balance needs to be maintained against perturbations while being adaptable in the presence of changes, a property known as robustness. Studies on the topological and functional properties of such networks have achieved some progress, but still have limited understanding of their robustness. Furthermore, more important a path is, higher is the chance to have a backup path. Thus, removing a link or an individual from any sub-network may also lead to blocking the information flow within that sub-network. The robustness of a model can also be assessed by means of altering the various parameters and components associated with forming a particular link. Robustness of a network can also be studied with respect to ‘resilience’, a method of analyzing the sensitivities of internal constituents under external perturbation, that may be random or targeted in nature [18].

## 1.7 Models of Infections

Basic disease models discuss the number of individuals in a population that are susceptible, infected and/or recovered from a particular infection. For this purpose, various differential equation based models have been used to simulate the events of action during the infection spread. In this scenario, various details of the infection progression are neglected, along with the difference in response between individuals. Models of infections can be categorized as SIR and Susceptible, Infected, Susceptible (SIS) [9].

### *1.7.1 Susceptible-Infected-Recovered (SIR)*

The SIR model considers individuals to have long-lasting immunity, and divides the population into those susceptible to the disease ($S$), infected ($I$) and recovered ($R$). Thus, the total number of individuals ($T$) considered in the population is

$$T = S + I + R \tag{1.11}$$

the transition rate from $S$ to $I$ is $\kappa$ and the recovery rate from $I$ to $R$ is $\rho$. Thus, the SIR model can be represented as

$$\frac{dS}{dT} = \gamma(T - S) - \kappa\frac{I}{T}S \tag{1.12}$$

$$\frac{dI}{dT} = \kappa\frac{I}{T}S - (\gamma + \rho)I \tag{1.13}$$

$$\frac{dR}{dT} = \rho I - \lambda R \tag{1.14}$$

Likewise, the reproductivity ($\theta$) of an infection can be identified as the average number of secondary instances a typical single infected instance will cause in a population with no immunity. It determines whether infections spreads through a population; if $\theta < 1$, the infection terminates in the long run; $\theta > 1$, the infection spreads in a population. Larger the value of $\theta$, more difficult is to control the epidemic [12].

Furthermore, the proportion of the population that needs to be immunized can be calculated by

$$\theta = \frac{\kappa}{\gamma + \rho} \tag{1.15}$$

Similarly, for $S(0)$, $I(0)$, $R(0)$, and $\theta <= 1$,

$$\lim_{t\to\infty}(S(t), I(t), R(t)) \to (T, 0, 0) \tag{1.16}$$

known as disease free stability, whereas if $\theta > 1$ and $I(0) > 0$, then

$$\lim_{t\to\infty} (S(t), I(t), R(t)) \to (\frac{T}{\theta}, \frac{\gamma T}{\kappa}(\theta - 1), \frac{\rho T}{\kappa}(\theta - 1)) \tag{1.17}$$

known as endemic stability can be identified. Depending upon these instances, immunization strategies can be initiated [6].

#### 1.7.1.1 Extensions to SIR Model

Although the contact network in a general SIR model can be arbitrarily complex, the infection dynamics can still being studied as well as modeled in a simple fashion. Contagion probabilities are set to a uniform value, i.e., $p$, and contagiousness has a kind of ‘on-off’ property, i.e., an individual is equally contagious for each of the $t_I$ steps while it has the infection, where 1 is present state of the system. One can extend the idea that contagion is more likely between certain pairs of individuals or vertices by assigning a separate probability $p_{v_i,v_j}$ to each pair of individuals or vertices $v_i$ and $v_j$, for which $v_i$ is linked to $v_j$ in a directed contact network.

Likewise, other extensions of the contact model involves separating the $I$ state into a sequence of early, middle, and late periods of the infection. For instance, it could be used to model an infection with a high contagious incubation period, followed by a less contagious period while symptoms are being expressed [16].

#### 1.7.1.2 Percolations of SIR Model

In most of the cases, SIR epidemics are thought of dynamic processes, in which the network state evolves step-by-step over time. It captures the temporal dynamics of the infection as it spreads through a population. The SIR model has been found to be suitable for infections, which provides lifelong immunity, like measles. In this case, a property termed as the force of infection is existent, a function of the number of infectious individuals is. It also contains information about the interactions between individuals that lead to the transmission of infection.

One can also have a static view of the epidemics where SIR model for $t_I = 1$. This means that considering a point in an SIR epidemic when a vertex $v_i$ has just become infectious, has one chance to infect $v_j$ (since $t_I = 1$), with probability $p$. One can visualize the outcome of this probabilistic process and also assume that for each edge in the contact network, a probability signifying the relationship is identified. Furthermore, one can also use the open and blocked healthy edges to represent the course of the infection spread. A vertex $v_i$ will become infected during the epidemic if and only if there is a path to $v_i$ from one of the initially infected nodes that consists entirely of open edges [3].

## 1.7.2 Susceptible-Infected-Susceptible (SIS)

The SIS model can be represented as

$$\frac{dS}{dT} = \rho I - \kappa S \tag{1.18}$$

$$\frac{dI}{dT} = \kappa S - \rho I \tag{1.19}$$

Removed state is absent in this case. Moreover, after a vertex is over with the Infectious state, it reverts back to the Susceptible state and is ready to initiate the infection again. Due to this alternation between the $S$ and $I$ states, the model is referred to as SIS model. The mechanics of SIS model can be discussed as follows [2].

1. At the initial stage, some vertices remain in $I$ state and all others are in $S$ state.
2. Each vertex $v_i$ that enters the $I$ state and remains infected for a certain number of steps $t_I$.
3. During each of these $t_I$ steps, $v_i$ has a probability $p$ of passing the infection to each of its susceptible directly linked neighbors.
4. After $t_I$ steps, $v_i$ no longer remains infected, and returns back to the $S$ state.

The SIS model is predominantly used for simulating and understanding the progress of STDs, where repeat infections are existent, like gonorrhoea. Moreover, certain assumptions with regard to random mixing between individuals within each pair of sub-networks are present. In this scenario, the number of neighbors for each individual is considerably smaller than the total population size. Such models generally avoid random-mixing assumptions thereby assigning each individual to a specific set of contacts that they can infect.

### 1.7.2.1 Life Cycle of SIS

An SIS epidemic, can run for long time duration as it can cycle through the vertices multiple number of times. If at any time during the SIS epidemic all vertices are simultaneously free of the infection, then the epidemic terminates forever. The reason being, no infected individuals exist that can pass the infection to others. In case if the network is finite in nature, a stage would arise when all attempts for further infection of healthy individuals would simultaneously fail for $t_I$ steps in a row.

Likewise, for contact networks where the structure is mathematically tractable, a particular critical value of the contagion probability $p$ is existent, an SIS epidemic undergoes a rapid shift from one that terminates out quickly to one that persists for a long time. In this case, the critical value of the contagion probability depends on the structure of the problem set [1].

## 1.8 Epidemic Invasions, Propagations and Outbursts

The patterns by which epidemics spread through vertex groups is determined by the properties of the pathogen, length of its infectious period, severity and the network structures. The path for an infection spread are given by a population state, with existence of direct contacts between the individuals or vertices. The functioning of network system depends on the nature of interaction between their individuals. This is essentially because of the effect of infection-causing individuals and topology of networks. To analyze the complexity of epidemics, it is important to understand the underlying principles of its distribution in the history of its existence. In recent years it has been seen that the study of disease dynamics in social networks is relevant with the spread of viruses and the nature of diseases [9].

Moreover, the pathogen and the network are closely intertwined with even within the same group of individuals, the contact networks for two different infections are different structures. This depends on respective modes of transmission of infections. For instance, a highly contagious infection, involving airborne transmission, the contact network includes a huge number of links, including any pair of individuals that are in contact with one another. Likewise, for an infection requiring close contact, the contact network is much sparser, with fewer pairs of individuals connected by links [7].

## 1.9 Combat and Immunization

Immunization is a site percolation problem where each immunized individual is considered to be a site which is removed from the infected network. Its aim is to transfer the percolation threshold that leads to minimization of the number of infected individuals. The model of SIR and immunization is regarded as a site-bond percolation model, and immunization is considered successful if the infected a network is below a predefined percolation threshold. Furthermore, immunizing randomly selected individuals requires targeting a large fraction, *frac*, of the entire population. For instance, some infections require 80–100 % immunization. Meanwhile, target-based immunization of the hubs requires global information about the network in question, rendering it impractical in many cases, which is very difficult in certain cases [6].

Likewise, social networks possess a broad distribution of the number of links, *conn*, connecting individuals and analyzing them illustrate that that a large fraction, *frac*, of the individuals need to be immunized before the integrity of the infected network is compromised. This is essentially true for scale-free networks, where $P(conn) \approx conn^{-\eta}$, $2 < \eta < 3$, where the network remains connected even after removal of most of its individuals or vertices. In this scenario, a random immunization strategy requires that most of the individuals need to be immunized before an epidemic is terminated [8].

For various infections, it may be difficult to reach a critical level of immunization for terminating the infection. In this case, each individual that is immunized is given immunity against the infection, but also provides protection to other healthy individuals within the population. Based on the SIR model, one can only achieve half of the critical immunization level which reduces the level of infection in the population by half. A crucial property of immunization is that these strategies are not perfect and being immunized does not always confer immunity. In this case, the critical threshold applies to a portion of the total population that needs to be immunized. For instance, if the immunization fails to generate immunity in a portion, *por*, of those immunized, then to achieve immunity one needs to immunize a portion

$$Im = \frac{\tau - 1}{\tau(1 - por)} \tag{1.20}$$

Here, *Im* denotes immunity strength. Thus, in case if *por* is huge it is difficult to remove infection using this strategy or provides partial immunity. It may also invoke in various manners: the immunization reduces the susceptibility of an individual to a particular infection, may reduce subsequent transmission if the individual becomes infected, or it may increase recovery.

Such immunization strategies require the immunized individuals to become infected and shift into a separate infected group, after which the critical immunization threshold ($S_I$) needs to be established. Thus, if *CIL* is the number of secondary infected individuals affected by an initial infectious individual, then

$$CIL = \frac{\tau - 1}{\tau - S_I} \tag{1.21}$$

Thus, $S_I$ needs to be less than one, else it is not possible to remove the infection. But, one also needs to note that an immunization works equally efficiently if it reduces the transmission or susceptibility and increases the recovery rate. Moreover, when the immunization strategy fails to generate any protection in a proportion *por* of those immunized, the rest $1 - por$ are fully protected. In this scenario, it can be not possible to remove the infection using random immunization. Thus, targeted immunization provides better protection than random-based [13].

### *1.9.1 Complex Topologies and Heterogeneous Structures*

In case of homogenous networks, the average degree, $\overline{conn}$, fluctuates less and can assume $conn \simeq \overline{conn}$, i.e., the number of links are approximately equal to average degree. However, networks can also be heterogeneous. Likewise, in a homogeneous network such as a random graph, $P(conn)$ decays faster exponentially whereas for heterogenous networks it decays as a power law for large *conn*.

The effect of heterogeneity on epidemic behavior studied in details for many years for scale-free networks. These studies are mainly concerned with the stationary limit and existence of an endemic phase. An essential result of this analysis is the expression of basic reproductive number which in this case is $\tau \infty \frac{\overline{conn^2}}{conn}$. Here, $\tau$ is proportional to the second moment of degree, which finally diverges for increasing network sizes [15].

## *1.9.2 Damage Patterns*

It has been noticed that the degree of interconnection in between individuals for all form of networks is quite unprecedented. Whereas, interconnection increases the spread of information in social networks, another exhaustively studied area contributes to the spread of infection throughout the healthy network. This rapid spreading is done due to less stringency of its passage through the network. Moreover, initial sickness nature and time of infection are unavailable most of the time, and the only available information is related to the evolution of the sick-reporting process. Thus, given complete knowledge of the network topology, the objective is to determine if the infection is an epidemic, or if individuals have become infected via an independent infection mechanism that is external to the network, and not propagated through the connected links.

If one considers a computer network undergoing cascading failures due to worm propagation whereas random failures due to misconfiguration independent of infected nodes, there are two possible causes of the sickness, namely, *random* and *infectious spread*. In case of *random* sickness, infection spreads randomly and uniformly over the network where the network plays no role in spreading the infection; and *infectious spread*, where the infection is caused through a contagion that spreads through the network, with individual nodes being infected by direct neighbors with a certain probability [6].

### 1.9.2.1 Random Sickness

In random damage, each individual becomes infected with an independent probability $\psi_1$. At time $t$, each infected individual reports damage with an independent probability $\psi_2$. Thus, on an average, a fraction $\psi$ of the network reports being infected, where

$$\psi = \psi_1.\psi_2 \tag{1.22}$$

It is already known that social networks possess a broad distribution of the number of links, $k$, originating from an individual. Computer networks, both physical and logical are also known to possess wide, scale-free, distributions. Studies of percolation on broad-scale networks display that a large fraction, $fc$, of the individuals need to be immunized before the integrity of the network is compromised. This is particularly true for scale-free networks, where the percolation threshold tends to 1, and the network remains contagious even after removal of most of its infected individuals [9].

#### 1.9.2.2 Infection Spread

When the hub individuals are targeted first, removal of just a fraction of these results in the breakdown of the network. This has led to the suggestion of targeted immunization of hubs. To implement this approach, the number for connections of each individual needs to be known. During infection spread, at time 0, a randomly selected individual in the network becomes infected. When a healthy individual becomes infected, a time is set for each outgoing link to an adjacent individual that is not infected, with expiration time exponentially distributed with unit average. Upon expiration of a link's time, the corresponding individual becomes infected, and in-turn begins infecting its neighbors [7].

### *1.9.3 Immunity*

In general, for an epidemic to occur in a susceptible population the basic reproductive rate must be greater than 1. In many circumstances not all contacts will be susceptible to infection. In this case, some contacts remain immune, due to prior infection which may have conferred life-long immunity, or due to some previous immunization. Therefore, not all individuals are infected and the average number of secondary infections decrease. Similarly, the epidemic threshold in this case is the number of susceptible individuals within a population that is required for an epidemic to occur. Similarly, the herd immunity is the proportion of population immune to a particular infection. If this is achieved due to immunization, then each case leads to a new case and the infection becomes more stable within the population [6].

One of the simplest immunization procedure consists of random introduction of immune individuals in the population for achieving uniform immunization density. In this case, for a fixed spreading rate, $\xi$, the relevant control parameter in the density of immune individuals present in the network, the immunity, *imm*. At the mean-field level, the presence of a uniform immunity reduces $\xi$ by a factor $1 - imm$, i.e., the probability of identifying and infecting a susceptible and non-immune individual becomes $\xi(1-imm)$. For homogeneous networks, one observes that, for aconstant $\xi$, the stationary prevalence is given by

$$\rho_{imm} = 0 \tag{1.23}$$

for $imm > imm_c$ and

$$\rho_{imm} = (imm_c - imm)/(1 - imm) \tag{1.24}$$

for $imm \leq imm_c$ Here $imm_c$ is the critical immunization value above which the density of infected individuals in the stationary state is null and depends on $\xi$ as $imm_c = 1 - \frac{\xi_c}{\xi}$.

Thus, for a uniform immunization level larger than $imm_c$, the network is completely protected and no large epidemic outbreaks are possible. On the contrary, uniform immunization strategies on scale-free heterogenous networks are totally ineffective. The presence of uniform immunization elocally depresses the infections prevalence for any value of $\xi$, and it is difficult to identify any critical fraction of immunized individuals that ensures the eradication of infection [2].

## 1.10 Understanding Cascading Failures, Natural Disturbances

Cascading, or epidemic processes are those where the actions, infections or failure of certain individuals increase the susceptibility of others. This results in the successive spread of infections from a small set of initially infected individuals to a larger set. Initially developed as a way to study human disease propagation, cascades ares useful models in a wide range of application. The vast majority of work on cascading processes focused on understanding how the graph structure of the network affects the spread of cascades. One can also focus on several critical issues for understanding the cascading features in network for which studying the architecture of the network is crucial [5].

The standard independent cascade epidemic model assumes that the network is directed graph $G = (V, E)$, for every directed edge between $v_i$, $v_j$, we say $v_i$ is a parent and $v_j$ is a child of the corresponding other vertex. Parent may infect child along an edge, but the reverse cannot happen. Let $V$ denote the set of parents of each vertex $v_i$, and for convenience $v_i \in V$ is included. Epidemics proceed in discrete time where all vertices are initially in the susceptible state. At time 0, each vertex independently becomes active, with probability $p_{init}$. This set of initially active vertices are called ‘seeds’. In each time step, the active vertices probabilistically infects its susceptible children; if vertex $v_i$ is active at time $t$, it infects each susceptible child $v_j$ with probability $p_{v_i v_j}$, independently. Correspondingly, a vertex $v_j$ susceptible at time $t$ becomes active in the next time step, i.e., at time $t + 1$, if any one of its parents infects it. Finally, a vertex remains active for only one time slot, after which it becomes inactive and does not spread the infection further as well as cannot be infected again either [5]. Thus, in this kind of an SIR epidemic, where some vertices remain forever susceptible because the epidemic never reaches them, while others transition, susceptible → active for one time step → inactive.

## 1.11 Conclusions

In this chapter, we discussed some critical issues regarding epidemics and their outbursts in static as well as dynamic network structures. We mainly focused on SIR and SIS models as well as identifying key strategies for identifying the damage caused in networks. We also discussed the various modeling techniques for studying cascading failures. Epidemics pass through populations and persists over long time periods. Thus, efficient modeling of the underlying network plays a crucial role in understanding the spread and prevention of an epidemic. Social, biological, and communication systems can be explained as complex networks with their degree distribution follows a power law, $P(conn) \approx conn^{-\eta}$, for the number of connections, *conn* for individuals, representing scale-free (SF) networks. We also discussed certain issues on epidemic spreading in SF networks characterized by complex topologies with basic epidemic models describing the proportion of individuals susceptible, infected and recovered from a particular disease. Likewise, we also explained the significance of the basic reproduction rate of an infection, that can be identified as the average number of secondary instances a typical single infected instance will cause in a population with no immunity. Also, we explained how determining the complete nature of a network required knowledge of every individual in a population and their relationships as, the problems with network definition and measurement depend on the assumptions of data collection processes. Nevertheless, we also illustrated the importance of invasion resistance methods, with temporary immunity generating oscillations in localized parts of the network, with certain patches following large numbers of infections in concentrated areas. Similarly, we also explained the significance of damages, namely, random, where the damage spreads randomly and uniformly over the network and in particular the network plays no role in spreading the damage; and infectious spread, where the damage spreads through the network, with one node infecting others with some probability.

## References

1. Anderson, R., May, R.: Infectious Diseases of Humans: Dynamics and Control. Oxford University Press, Oxford (1991)
2. Bailey, N.: The Mathematical Theory of Infectious Diseases and its Applications. Hafner Press, New York (1975)
3. Bak, P., Chen, K., Tang, C.: A forest-fire model and some thoughts on turbulence. Phys. Lett. A. **147**, 297–300 (1990)
4. Barabási, A., Albert, R.: Emergence of scaling in random networks. Science **286**(5439), 509–512 (1999)
5. Casals, M., Guzman, K., Cayla, J.: Mathematical models used in the study of infectious diseases. Rev. Esp. Salud. Publica. **83**(5), 689–95 (2009)
6. Ellis, L.: Spread of Epidemic Disease on Networks. Wiley, New York (2001)
7. Keeling, M., Eames, K.: Networks and epidemic models. J. R. Soc. Interf. **2**(4), 295–307 (2005)
8. Kenah, E., Robins, J.: Network-based analysis of stochastic sir epidemic models with random and proportionate mixing. J. Theor. Biol. **249**(4), 706–722 (2007)
9. Kot, M.: Elements of Mathematical Ecology. Cambridge University Press, Cambridge (2001)
10. Król, D.: On modelling social propagation phenomenon. In: Nguyen, N., Attachoo, B., Trawiński, B., Somboonviwat, K. (eds.) Intelligent Information and Database Systems. Lecture Notes in Computer Science, pp. 227–236 (2014)
11. Król, D.: Propagation phenomenon in complex networks: theory and practice. New Gener. Comput. **32**(3–4), 187–192 (2014)

12. Lotka, A.: Relation between birth rates and death rates. Science **26**, 121–130 (1907)
13. MacDonald, G.: The analysis of malaria epidemics. Trop. Dis. Bull. **50**(10), 871–889 (1953)
14. Mason, O., Verwoerd, M.: Graph theory and networks in biology. IET Syst. Biol. **1**(2), 89–119 (2007)
15. Murray, J.: Mathematical Biology. Springer, New York (2002)
16. Newman, M.: Spread of epidemic disease on networks. Phys. Rev. E. Stat. Nonlin. Soft Matter Phys. **66**(1), 0168,128 (2002)
17. Nokes, D., Anderson, R.: The use of mathematical models in the epidemiology study of infectious diseases and in the design of mass vaccination programmes. Epidemiol. Infect **101**, 1–20 (1988)
18. Rhodes, C., Anderson, R.: Forest-fire as a model for the dynamics of disease epidemics. J. Frankl. Inst. **335**(2), 199–211 (1998)
19. Rhodes, C., Jensen, H., Anderson, R.: On the critical behaviour of simple epidemics. Proc. Biol. Sci. **264**(1388), 1639–46 (1997)
20. Sobol, I.: Sensitivity estimates for nonliner mathematical models. Math. Model. Comput. Exp. **1**(12), 577–596 (1993)
21. Tran, L., Rizk, M., Liao, J.: Ensemble modeling of metabolic networks. Biophys. J. **95**(12), 5606–5617 (2008)
22. Trapman, P.: On analytical approaches to epidemics on networks. Theor. Popul. Biol. **71**(2), 160–73 (2007)
23. Vallabhajosyula, R., Raval, A.: Computational modeling in systems biology. Methods Mol. Biol. **662**(5), 97–120 (2010)
24. Watts, D., Strogatz, S.: Collective dynamics of ‘small-world’ networks. Nature **393**(6684), 440–442 (1998)
25. Zinck, R., Grimm, V.: Unifying wildfire models from ecology and statistical physics. Am. Nat. **174**, E170–E185 (2009)

# Chapter 2
# Information Propagation in a Social Network: The Case of a Fish Schooling Algorithm

**A. Brabazon, W. Cui and M. O'Neill**

**Abstract** The propagation of information about the environment amongst animals via social communication has attracted increasing research interest in recent decades with the realisation that many animal species engage in subtle forms of information transfer which had previously escaped notice. From an evolutionary perspective, the widespread existence of social communication mechanisms is not surprising given the significant benefits which can accrue to behaviours such as sharing of information on resources and on environmental threats. More generally, we can consider this process as information flowing between a network of nodes or agents, wherein each agent receives inputs from their senses, processes this information, and in turn through their resulting actions, can influence subsequent actions of other agents. Social communication mechanisms of organisms have inspired the development of several powerful families of optimization algorithms including ant colony optimization and honey bee optimization algorithms. One interesting example of information propagation is provided by the shoaling and schooling behaviours of fish. In this chapter we develop an optimization algorithm (the *Fish Algorithm*) which is inspired by the schooling behaviour of 'golden shiner' fish (*Notemigonus crysoleucas*) and explore the relative importance of social information propagation and individual perception mechanisms in explaining the resulting performance of the algorithm.

A. Brabazon (✉) · W. Cui · M. O'Neill
Complex Adaptive Systems Laboratory and School of Business,
University College Dublin, Dublin, Ireland
e-mail: anthony.brabazon@ucd.ie

W. Cui
e-mail: wei.cui.ireland@gmail.com

M. O'Neill
e-mail: m.oneill@ucd.ie

D. Król et al. (eds.), *Propagation Phenomena in Real World Networks*,
Intelligent Systems Reference Library 85, DOI 10.1007/978-3-319-15916-4_2

## 2.1 Introduction

Swarm behaviour has long attracted research attention with the ‘flocking’ (‘boids’) simulation by Reynolds [31], which mimicked the flocking behaviour of birds, being one of the earliest and best-known examples of such work. In these simulations the flock has no leader (no global control) and co-ordinated movement emerges from the local interactions of individuals in the population. The simulation embeds a few simple rules whereby individuals move in the same direction as their neighbours, remain close to their neighbours, and avoid collisions with their neighbours (producing *alignment, cohesion and separation*). The key characteristic is that each agent only needs local information when deciding how to adjust their movements and yet this, allied to the three simple rules, is sufficient to ensure globally-coordinated behaviour at flock level.

More recently, mechanisms of collective intelligence and their application as practical problem-solving tools, has attracted considerable research interest leading to the development of several families of swarm-inspired algorithms including, ant-colony optimization [7, 9–12], particle swarm optimization [13, 17, 18], bacterial foraging optimization algorithms [28, 29], honey bee algorithms [8, 23, 30, 41], and a developing literature on fish school algorithms. A critical aspect of all of these algorithms is that powerful, emergent, problem-solving occurs as a result of the propagation or sharing of information among a network of individuals, where each individual only possesses local information. Typically the algorithms emphasise the importance of sensing and of communication processes between the agents, and this leads in turn to a discussion of what the agents ‘know’ and how information is propagated or ‘spread’ between individual nodes or agents in the population.

### *2.1.1 Fish Schooling*

Biologists draw an important distinction between dispersion and aggregation economies. In a dispersion economy an increase in group size is correlated with a decrease in the fitness of individual group members, so maximal welfare is obtained when individuals are dispersed and solitary. In contrast, aggregation economies emphasise how group membership can increase the survival rate of individuals particularly when population density is low. A particular example of an aggregation economy is exhibited by some social species of fish which ‘shoal’. ‘Shoaling behaviour’ occurs when fish are observed to cluster together. If the fish also demonstrate a tendency to swim in the same direction in a coordinated manner they are said to ‘school’. These behaviours are common. Approximately a quarter of fish shoal for their entire lives (‘obligate shoalers’ such as tuna, herrings and anchovy) and approximately half shoal for at least part of their lives (‘facultative shoalers’ such as Atlantic cod). More than 4,000 species of pelagic fish are known to be schooling [33] and fish aggregations can be very large with Parrish et al. [27] noting that herring can form schools of a billion or more fish.

Fish shoal and school for mutual protection and to synergistically achieve certain tasks [6]. The benefits include defence against predators as the shoal possesses 'many eyes' (or distributed sensing) and has a high-level of vigilance. There is also better protection from individual capture by predators due to the *predator confusion effect* (the many moving targets overloads the predator's visual channel). The shoal may exhibit enhanced foraging success as many eyes search for food and information on food finds is transmitted through the shoal as the fish can visually monitor each other's behaviour. Another claimed benefit of schooling is increased hydrodynamic efficiency as the school moves through the water [33].

Schooling behaviour may also reduce or even eliminate the need for sleep. During waking, the brain of most vertebrates is busy processing sensory information, particularly visual information, and this conflicts with the need to refresh and consolidate memories [22]. During schooling, the need for sensory processing, particularly by fish inside the school, is greatly lessened and the burden of sensory processing is shifted from individuals to the entire school [22]. Schooling behaviours may therefore play a role similar to that of restful waking or sleep in non-schooling fish species.

### 2.1.2 How Do Fish Schools Make Decisions?

A natural question facing any modeller who is seeking to develop an optimization algorithm using fish school inspired behaviours is how do fish schools actually make decisions—and critically, is there any theoretical reason to suppose that distributed sensing can generate a more 'intelligent' decision than the decision that could be made by an individual fish?

When we consider the dynamic environment which faces a school of fish, it is apparent that many complex decisions are faced. In which direction should it swim if faced by a predator? When should it stop and forage? When and where should it migrate? In contrast to mammal herds, fish schools have no leader. Each of the fish in a school has similar sensing capabilities and similar behaviour patterns for acting on sensory information [33] but there is no strong evidence that individual fish can undertake highly complex information processing.

A recent study [35] has suggested that fish schools may implement a form of consensus-based decision making employing a simple quorum rule. Under a quorum rule, an individual's probability of committing to a particular decision option increases sharply when a threshold number of other individuals have committed to it. Hence, if individuals can observe the decisions of others before committing themselves to a decision such as what direction in which to swim, a relatively naive copying behaviour can be an effective strategy for successful decision making, without the need for individuals to undertake complicated information processing.

Distributed perception and quorum decision processes combine therefore to create a form of collective intelligence which can reduce the need to undertake complex cognition at agent level, and can also allow robust decision making to take place even when individual perceptions are noisy. The quality of the decision and the size of the group are highly correlated [35] so the quality of the decision increases as group size increases. This suggests that fish school behaviours can indeed form a useful platform for the development of optimization algorithms. In this study we propose an optimization algorithm inspired by a recent study by Berdahl et al. [3] of golden shiner fish and within this framework, explore the relative importance of social information propagation and individual perception mechanisms in explaining the resulting performance of the algorithm.

The remainder of this contribution is organised as follows. Section 2.2 provides some background literature on previous work which has adopted a fish school metaphor in the development of optimization algorithms and on the specific biological model underlying this study. Section 2.3 describes the proposed algorithm (termed the 'Fish Algorithm'). The results from a series of test problems are provided in Sect. 2.5 and finally, conclusions and opportunities for future work are discussed in Sect. 2.6.

## 2.2 Background

A number of previous studies have previously employed a fish school metaphor to develop algorithms for optimization and clustering ([1, 2, 16, 19, 38, 44] provide a sampling of this work). Two of the better-known approaches are Fish School Search (FSS) [2] and the Artificial Fish Swarm Algorithm (AFSA) [19].

In FSS the algorithm implements three fish behaviours, namely feeding, swimming and breeding. The behaviour of feeding is inspired by the natural instinct of fishes to feed, feeding here is a metaphor for the evaluation of candidate solutions in the search space; the swimming behaviour aims at mimicking the coordinated movement of fish in a school guiding the search process; the breeding behaviour is inspired by natural selection a metaphor for exploitation of better-adapted candidate solutions. The fish (agents) swim (search) for food (candidate solutions) in an aquarium (search space) and the weight of each fish acts as an innate memory of its past individual success. Unlike particle swarm optimization (PSO) [17, 18], no direct memory of a personal best location or a global best location is maintained. FSS has shown itself to be a powerful optimization algorithm demonstrating good results on a range of optimization problems.

The AFSA [19] embeds a number of fish behaviours including preying, swarming, and following so that the behaviour of an artificial fish depends on its its current state, its local environmental state (including the quality of its current location and the states of nearby companions). A good review of the recent literature on AFSA is provided in [24].

### *2.2.1 Application of Fish School Algorithms*

Fish school algorithms have been applied for a wide variety of applications and an excellent overview of these is provided by [24]. Canonical versions of fish school algorithms typically employ a real-valued representation and are used to search in an environment/problem space for a ‘point’ which corresponds to an optimal solution vector (a simple exemplar would be a vector of parameters for a mathematical model which is being calibrated using a training dataset). Hence, the algorithms can be applied to any real-valued optimization problem. The canonical algorithms can also be modified for application to discrete optimization, multi-objective optimization and clustering.

A sampling of the applications for which fish school algorithms have been employed include, the determination of the optimal deployment strategy for nodes in a wireless network [4, 42]; the optimal deployment of directional visible light sensor networks for battlefield surveillance and intrusion detection [43]; road traffic network design [21]; the optimization of weights in a feed-forward neural network model [40]; quality of service (QoS) graded optimization in electric power communication networks [25]; the optimization of the parameters of membership functions for a fuzzy logic controller [36]; task scheduling in a multi robot group [37]; aircraft landing scheduling in a multi-runway airport [5]; and efficient job scheduling in grid computing [14].

### *2.2.2 Golden Shiner Fish*

A practical issue that arises in attempting to develop an algorithm based on the behaviour of fish schools is that we have relatively little hard data on the behavioural mechanisms which underlie schooling phenomena. At the level of the individual, agents respond to their own sensory inputs, physiological and cognitive states, and locomotory constraints [15] and it is not trivial to disentangle the relative influence of each of these. At group-level, it is often difficult to experimentally observe the mechanics of the movement of animal groups or fish schools, and hence much previous work developing fish school algorithms has relied on high-level observations of fish behaviour rather than on granular empirical data on these behaviours.

In this study we draw inspiration from a detailed study of the behaviour of a species of schooling fish ‘golden shiners’ which display a marked preference for shaded habitat [3]. These fish are strongly social and form shoals of some 200–250 individuals in the fresh-water lakes where they live.

In order to investigate the mechanism underlying the observed collective response of golden shiner fish to light gradients, fish were tracked individually to obtain information on individual and group trajectories. The study examined the degree to which the motion of individuals is explained by individual perception (steepest direction of light gradient as seen by the individual fish) and social influences based on distributed perception (positions of conspecifics). The results indicated that an individual’s acceleration was more influenced by the location of conspecifics than by locally-perceived environmental gradients. When the magnitude of the social vector was high (all conspecifics moving in similar direction) the social influence was dominant. As noted by [32], all forms of animal communication are closely tied to the senses. In the case of fish, visual cues form the primary basis of the social communication mechanism as schooling fish are able to observe the movements of their neighbouring conspecifics.

## 2.3 Fish Algorithm

An important question that underlies the design of foraging strategies, or the design of optimization algorithms, is what is the most effective way of searching for objects whose location is not known a priori. In foraging, the search could be guided by external cues, either via past experience (memory) or sensory inputs (such as vision) of the searcher. Alternatively, the search process could be stochastic (i.e. undirected). When the location of the target objects is unknown, a degree of ‘guessing’ is unavoidable, and probabilistic or stochastic strategies are required [39].

In the proposed algorithm, the movement of each fish is governed by three biologically-inspired factors which are described below, and also embeds a stochastic element. In each iteration of the algorithm, a fish is displaced from its previous position through the application of a velocity vector:

$$p_{i,t} = p_{i,t-1} + v_{i,t} \tag{2.1}$$

where $p_{i,t}$ is the position of the $i$th fish at current iteration of the algorithm ($t$), $p_{i,t-1}$ is the position of the $i$th fish at the previous iteration ($t-1$), and $v_{i,t}$ is its velocity.

The velocity update is a composite of three elements, prior period velocity, an individual perception mechanism, and social influence via the distributed perception of conspecifics. The update is:

$$v_{i,t} = v_{i,t-1} + DP_{i,t} + IP_{i,t} \tag{2.2}$$

or more generally

$$v_{i,t} = w_1 v_{i,t-1} + w_2 DP_{i,t} + w_3 IP_{i,t} \tag{2.3}$$

The difference between the two update equations is that weight coefficients are given to each of the update items in Eq. 2.3. In all the experiments of this study, Eq. 2.2 is used for velocity update. While the form of the velocity update bears a passing resemblance to the standard PSO velocity update, in that both have three terms, it should be noted that the operationalisation of the individual perception and distributed perception mechanisms is completely different to the memory-based concepts of *pbest* and *gbest* in PSO. The next subsection explains the operation of the two perception mechanisms.

### *2.3.1 Prior Period Velocity*

The inclusion of a prior period velocity can be considered as a proxy for momentum or inertia. Although this feature was not described in the study of golden shiner fish [3], the inclusion of this term is motivated by empirical evidence from the movement ecology literature which indicates that organisms tend to move with a 'directional persistence' [39].

### 2.3.2 Distributed Perception Influence

In all social models, a key element is how the overall population influences the decisions of each agent at each time step. Typically, the actions of each agent are influenced by a subset of the population who are within an 'interaction range' of them. This influence can be modelled in a variety of ways including the fraction of an individual's neighbours taking a particular course of action or the action of their nearest neighbour. In this study we model the distributed perception influence for the $i$th fish by the following:

$$DP_i = \frac{\sum_{j=1}^{N_i^{DP}} (p_j - p_i)}{N_i^{DP}}, \qquad j \neq i \tag{2.4}$$

where $p_i$ is the position of the $i$th fish, and the sum is calculated over all neighbours within an assumed range of interaction of the $i$th fish $r_{DP}$, that is $0 <| p_j - p_i |\leq r_{DP}$, where $p_j$ is the position of the $j$th neighbouring fish, and $N_i^{DP}$ is number of neighbours in the assumed range of interaction of the $i$th fish. If there are no neighbours in its assumed range of interaction, this term becomes zero. Figure 2.1 shows how the $i$th fish is affected by the three neighbouring fish ($p_1$, $p_2$, $p_3$) which are within its visible range (defined by the radius $r_{DP}$).

Alternative methods of modelling this social influence could be implemented such as only considering neighbours within the angular visual range of each agent as suggested by Miller et al. [26]. While this would be more plausible from a biological perspective, it would impose additional computational complexity so we use a simpler approach in this chapter which implicitly assumes 360° vision. Note that

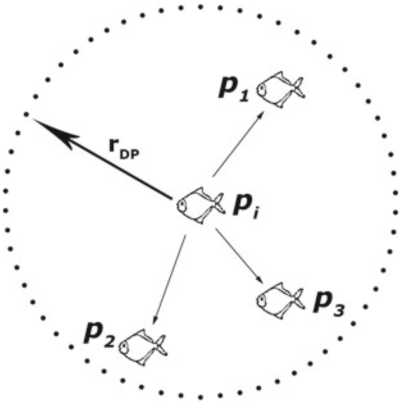


**Fig. 2.1** Illustration of distributed perception

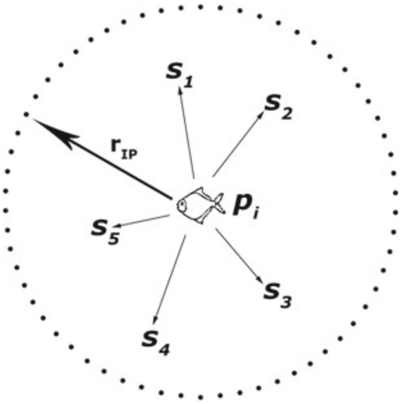


**Fig. 2.2** Illustration of individual perception

in this mechanism, no direct account is taken of the light gradient in any direction by an individual fish, rather the influence on the movement of a fish is completely determined by the movement of its neighbours.

### *2.3.3 Individual Perception Influence*

Individual perception is implemented as follows. At each update, each fish assesses the local 'light' gradient surrounding it, by drawing $N_i^{IP}$ samples within an assumed 'visibility' region of radius $r_{IP}$. While a real-world fish will have a specific angle of vision depending on its own body structure, we adopt a random sampling in a hypersphere around the fish on grounds of generality. The individual perception influence for the $i$th fish is determined by:

$$IP_i = \frac{\sum_{j=1}^{N_i^{IP}} (s_j - p_i) * fit_j}{\sum_{j=1}^{N_i^{IP}} fit_j} , \qquad j \neq i \tag{2.5}$$

where $p_i$ is the position of the $i$th fish, $r_{IP}$ is the radius of the assumed range within which the $i$th fish can sense environmental information, $N_i^{IP}$ is the number of samples which the $i$th fish generates, $s_j$ is the position of the $j$th sample $(0 <\mid s_j - p_i \mid \leq r_{IP})$, and $fit_j$ is the fitness value (or 'quality') of the $j$th sample. Figure 2.2 demonstrates how the $i$th fish is influenced by the five random samples $(s_1 - s_5)$ in the perception range with a radius $r_{IP}$.

## 2.4 Experimental Design

In this section we describe the test functions used in all our experiments, we outline the precise experiments undertaken in this study, and we describe the associated experimental parameters.

### *2.4.1 Benchmark Functions*

Twelve standard benchmark problems (outlined in Table 2.1) taken from the optimization literature were used to test the developed algorithms. All problems are examined at two levels of dimensionality, namely 30 and 60 dimensions. The aim in all the experiments is to find the vector of values which minimises the value of a test function, hence, we can define the fitness of a solution vector as the value of the test function at that location, with lower values (in this case, as we are minimising) indicating a better quality (or ‘fitter’) solution.

Two of the functions namely, the Sphere and Rosenbrock functions, represent unimodal problems. The Griewank and Rastrigin functions are more complex and contain multiple local optima. In following paragraphs, we provide a brief description of these test functions in order to provide some intuition as to their structure.

The last six problems are drawn from the optimization benchmark functions used in the IEEE CEC 2005 Special Session on Real-Parameter Optimization [34]. An interesting aspect of these functions is that the global optima are shifted or rotated (shift is given by the parameter $\boldsymbol{o}$, and the parameter $M$ represents an orthogonal matrix which is used to rotate the function). The net effect of these processes is to

**Table 2.1** Twelve optimization problems

| Name | Function | Search space | Optima |
|---|---|---|---|
| Sphere | $F_1(\mathbf{x}) = \sum_{i=1}^{n} x_i^2$ | $[-3.12\ 7.12]^D$ | 0 |
| Rosenbrock | $F_2(\mathbf{x}) = \sum_{i=1}^{n-1}[100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2]$ | $[-30\ 30]^D$ | 0 |
| Ackley | $F_3(\mathbf{x}) = -20 \exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}\right)$<br>$- \exp\left(\frac{1}{D}\sqrt{\sum_{i=1}^{D} \cos(2\pi x_i)}\right) + 20 + e$ | $[-32.768\ 32.768]^D$ | 0 |
| Griewank | $F_4(\mathbf{x}) = 1 + \sum_{i=1}^{n} \frac{x_i^2}{4000} - \prod_{i=1}^{n} \cos(\frac{x_i}{\sqrt{i}})$ | $[-600\ 600]^D$ | 0 |
| Rastrigin | $F_5(\mathbf{x}) = 10n + \sum_{i=1}^{n}[x_i^2 - 10\cos(2\pi x_i)]$ | $[-5.12\ 5.12]^D$ | 0 |
| Schwefel | $F_6(\mathbf{x}) = 418.9829 \times D - \sum_{i=1}^{D} x_i \sin\left(\lvert x_i\rvert^{\frac{1}{2}}\right)$ | $[-500\ 500]^D$ | 0 |
| Shifted sphere | $F_7(\mathbf{x}) = \sum_{i=1}^{D} z_i^2 - 450\ ,\ \mathbf{z} = \mathbf{x} - \mathbf{o}$ | $[-100\ 100]^D$ | −450 |
| Shifted rosenbrock | $F_8(\mathbf{x}) =$<br>$\sum_{i=1}^{D-1} 100(z_i^2 - z_{i+1})^2 + (x_i - 1)^2 + 390\ ,$<br>$\mathbf{z} = \mathbf{x} - \mathbf{o} + 1$ | $[-100\ 100]^D$ | 390 |
| Shifted rotated ackley | $F_9(\mathbf{x}) = -20\exp(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} z_i^2})$<br>$- \exp(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi z_i))$<br>$+ 20 + e - 140\ ,\ \mathbf{z} = (\mathbf{x} - \mathbf{o}) * M$ | $[-32\ 32]^D$ | −140 |
| Shifted rotated griewank | $F_{10}(\mathbf{x}) = \sum_{i=1}^{D} \frac{z_i^2}{4000} - \prod_{i=1}^{D}\cos(\frac{z_i}{\sqrt{i}}) + 1$<br>$- 180\ ,\ \mathbf{z} = (\mathbf{x} - \mathbf{o}) * M$ | $[-600\ 600]^D$ | −180 |
| Shifted rotated rastrigin | $F_{11}(\mathbf{x}) = \sum_{i=1}^{D} (z_i^2 - 10\cos(2\pi z_i) + 10)$<br>$- 330\ ,\ \mathbf{z} = (\mathbf{x} - \mathbf{o}) * M$ | $[-5\ 5]^D$ | −310 |
| Shifted schwefel | $F_{12}(\mathbf{x}) = \sum_{i=1}^{D}(\sum_{j=1}^{i} z_i)^2 - 450\ ,\ \mathbf{z} = \mathbf{x} - \mathbf{o}$ | $[-100\ 100]^D$ | −450 |

move the global optimum away from the origin in each case, due to the known issues with using standard, benchmark functions which have their optimum at the origin [20]. These issues can sometimes be exploited by algorithms to produce an upward bias in reported performance. Problems include the fact that,

1. many popular benchmark functions are symmetric, and hence have the same optimal parameter values for all dimensions (for example, a vector of zeros); and
2. the global optimum may lie at the centre of the search space (this can produce problems if search agents are initialised randomly along the range of each dimension).

Hence, considering the conventional sphere function,

$$f(x) = \sum_{i=1}^{D} x_i^2$$

the shifted sphere function is given by:

$$f(x) = \sum_{i=1}^{D} (x_i - o_i)^2$$

and the shifted rotated sphere function is given by:

$$f(x) = \sum_{i=1}^{D} [(x_i - o_i) * M]^2$$

#### 2.4.1.1 Sphere Function

This is a relatively simple test function as it is continuous, convex and unimodal. The function is defined as $\sum_{i=1}^{n} x_i^2$. In Fig. 2.3, $n$ is set to 2 for ease of illustration, and $-5.12 \leq x_i \leq 5.12$. The objective is to find the values of $x_1$ and $x_2$ which minimise the value of the function. By inspection, the global minimum (zero) occurs when $x_1$ and $x_2$ are zero. While we illustrate the function here for the case where there are two inputs, in our experiments on each test function we undertake a search for the global optimum in both 30 and 60 dimensions.

#### 2.4.1.2 Griewangk's Function

Griewangk's function has many local minima in the region of the global minimum, with these minima being regularly distributed. The presence of many local minima renders the determination of the optimal value for this function more difficult than is the case for the Sphere function. The function is defined as:

$$F(x) = 1 + \sum_{i=1}^{n} \left[ \frac{x_i^2}{4000} \right] - \prod_{i=1}^{n} \left[ \cos\left( \frac{x_i}{\sqrt{i}} \right) \right] \tag{2.6}$$

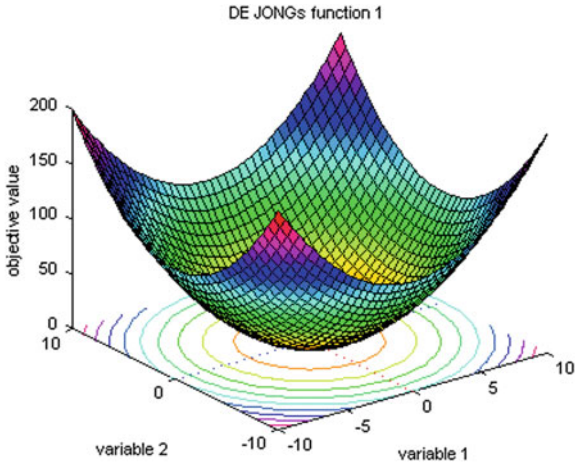


**Fig. 2.3** Sphere function

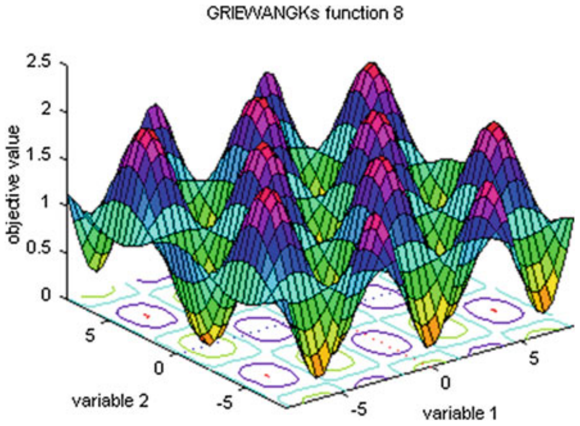


**Fig. 2.4** Griewangk's function, range +/− 5

where $n = 2$ (in Fig. 2.4), and $-600 \le x_i \le 600$. The global minimum (zero) occurs when all $x_i$ are 0.

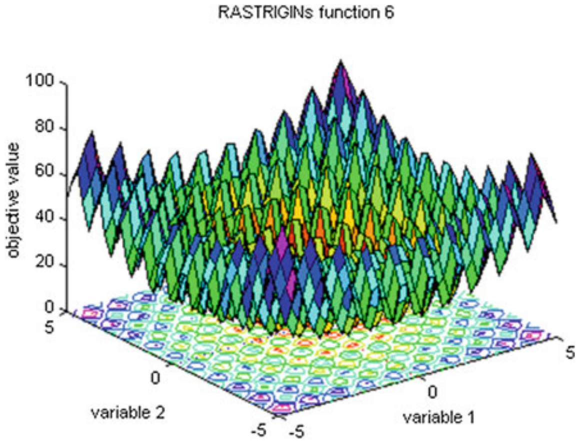
### 2.4.1.3 Rastrigin's Function

Rastrigin's function has a cosine modulation to produce many local minima (Fig. 2.5). This produces a test function which is highly multimodal. However, the location of the minima are regularly distributed. The function is defined as:

$$F(x) = n * A + \sum_{i=1}^{n} \left[ x_i^2 - A * \cos(2\pi x_i) \right] \tag{2.7}$$

with $A = 10$ and $n = 2$ (in the illustration ) and $-5.12 \le x_i \le 5.12$. The global minimum (zero) occurs when all $x_i$ are zero.

**Fig. 2.5** Rastrigin's function



The Rosenbrock function (also known as Rosenbrock's valley or Rosenbrock's banana function) is a non-convex function. The global minimum is inside a long, narrow, parabolic shaped flat valley. While it is relatively easy to find the valley, it is difficult to find the global optimum point within this.

### *2.4.2 Experiments*

Two groups of experiments are undertaken. Initially, we determine the performance of the canonical fish algorithm (denoted as '*FA*') which uses the velocity update described in Eq. 2.2, on all the test problems. Next we develop three variants of the canonical FA which switch off, in turn, the momentum, the distributed perception (DP) and the individual perception (IP) influences (these algorithmic variants are denoted as FA1, FA2 and FA3 respectively). The performance of each of these variants on the test problems is examined in order to gain insight into the role that each of the three components of the velocity update step plays in determining the FA's overall performance.

The second set of experiments examines the sensitivity of the canonical FA to changes in two of its parameters, namely the radius of perception in both $r_{DP}$ &$r_{IP}$, and the number of samples (denoted as $s$) used in the simulated individual perception (IP) component. The chosen values of these parameters are shown in Table 2.2.

From a biological point of view, it is plausible to assume that fish have a bigger radius for DP than IP, namely $r_{DP} > r_{IP}$. The value chosen for the two radii is problem specific, as it is influenced by the choice of the number of fish ($N$), the radius (size) of the search space ($R$) and the dimensionality of the this space ($D$). In the FA algorithm, the values of $r_{DP}$ and $r_{IP}$ were chosen after initial experimentation as $\frac{R}{1.5\sqrt[D]{N}}$ and $\frac{R}{1.8\sqrt[D]{N}}$ so that in most cases each fish has neighbouring fish within the radius $r_{DP}$.

**Table 2.2** Parameter setting of algorithms

| Algorithm | Radius of DP ($r_{DP}$) | Radius of IP ($r_{IP}$) | Number of samples in IP ($s$) | Velocity updating equation |
|---|---|---|---|---|
| FA | $\frac{R}{1.5\sqrt[D]{N}}$ | $\frac{R}{1.8\sqrt[D]{N}}$ | 5 | $v_{i,t} = v_{i,t-1} + DP_{i,t} + IP_{i,t}$ |
| FA1 | $\frac{R}{1.5\sqrt[D]{N}}$ | $\frac{R}{1.8\sqrt[D]{N}}$ | 5 | $v_{i,t} = 0 + DP_{i,t} + IP_{i,t}$ |
| FA2 | $\frac{R}{1.5\sqrt[D]{N}}$ | $\frac{R}{1.8\sqrt[D]{N}}$ | 5 | $v_{i,t} = v_{i,t-1} + 0 + IP_{i,t}$ |
| FA3 | $\frac{R}{1.5\sqrt[D]{N}}$ | $\frac{R}{1.8\sqrt[D]{N}}$ | 5 | $v_{i,t} = v_{i,t-1} + DP_{i,t} + 0$ |
| FAa | $\frac{R}{3\sqrt[D]{N}}$ | $\frac{R}{3.6\sqrt[D]{N}}$ | 5 | $v_{i,t} = v_{i,t-1} + DP_{i,t} + IP_{i,t}$ |
| FAb | $\frac{R}{1\sqrt[D]{N}}$ | $\frac{R}{1\sqrt[D]{N}}$ | 5 | $v_{i,t} = v_{i,t-1} + DP_{i,t} + IP_{i,t}$ |
| FAc | $\frac{R}{1.5\sqrt[D]{N}}$ | $\frac{R}{1.8\sqrt[D]{N}}$ | 10 | $v_{i,t} = v_{i,t-1} + DP_{i,t} + IP_{i,t}$ |
| FAd | $\frac{R}{1.5\sqrt[D]{N}}$ | $\frac{R}{1.8\sqrt[D]{N}}$ | 1 | $v_{i,t} = v_{i,t-1} + DP_{i,t} + IP_{i,t}$ |

*Note* R is the radius of the search space
D is the dimension of the test problem
N is the number of fish

In order to undertake some sensitivity analysis, four variants of the FA algorithm are developed. In the FAa algorithm, the values of $r_{DP}$ and $r_{IP}$ are set to be half of those in the FA algorithm. In the FAb algorithm, the values of $r_{DP}$ and $r_{IP}$ are set to be larger than those in the FA algorithm. In the FAc algorithm, the value of $s$ is increased to 10 (as against 5 in the FA algorithm). In the FAd algorithm, the value of $s$ is reduced to 1. Note that in these latter two cases, the effect is to alter the implicit weighting accorded to the IP mechanism in the velocity update step, as in all our experiments, each algorithmic variant is accorded the same number of function evaluations.

We note that in this study the focus is not on designing the 'best' possible variant of the fish algorithm for optimization purposes. Rather, using the framework outlined in Sect. 2.3 we seek to examine the relative importance of social information propagation and individual perception mechanisms in explaining the resulting performance of the algorithm. We also wish to examine the sensitivity of the performance to changes in key parameters in each mechanism (range of perception and relative weight placed on IP vs. DP).

### 2.4.3 Experimental Settings

Table 2.3 describes the parameter settings adopted. In each experiment, 40 fish are used. All reported results are averaged over 30 runs and we test the statistical significance of all differences in the means using a $t$-test. In all experiments, an equivalent number of function evaluations are undertaken in order to ensure a fair comparison between the different algorithms. The experiments were undertaken on an Intel Core i7 (2.93 GHz) system with 12 GB RAM.

**Table 2.3** Parameter setting of experiments

| Parameters | Values |
|---|---|
| Trials | 30 |
| Size of fish school | $N = 40$ |
| Dimension of problem | $D = 30, 60$ |

## 2.5 Results

Tables 2.4, 2.5, 2.6 and 2.7, and Figs. 2.6 and 2.7 present the results from our experiments. The Tables show for each algorithm variant & test function combination (for both D = 30 and D = 60), the end of run evaluation for each test function at the best location (solution vector) found across all 30 runs ('Best'), the evaluation of each benchmark function averaged over the best location (solution vector) found on each of the 30 individual runs ('Mean'), and the associated standard deviation over all 30 runs. The Tables also present the results from our statistical testing of a variety of hypotheses. In all cases, low $p$ values indicate that the null hypothesis of 'no difference between the means' is rejected (a 95 % level is applied).

Figures 2.6 and 2.7 illustrate the 'Mean' (defined as above) evaluation of each benchmark function and indicate how this value changes (improves) as the number of iterations increases (only the D = 60 case is shown in order to conserve space).

### 2.5.1 *Hypotheses Examined*

In order to facilitate interpretation of the statistical tests we outline the notation used below.

The first set of hypotheses concern the testing of the importance of each component of the fish algorithm (FA). The null hypothesis is that there is no difference in the performance (i.e. 'Mean') between the algorithm with a component turned off and the canonical FA. Therefore three hypotheses are tested as follows.

- $H_1$: no difference in performance between the FA and the FA1 algorithm;
- $H_2$: no difference in performance between the FA and the FA2 algorithm;
- $H_3$: no difference in performance between the FA and the FA3 algorithm.

The next set of hypotheses concern the analysis of differing parameter settings for FA. Four cases are examined, FAa, FAb FAc and FAd and the relevant hypotheses are denoted as follows.

- $H_a$: no difference in performance between the FA and the FAa algorithm;
- $H_b$: no difference in performance between the FA and the FAb algorithm;
- $H_c$: no difference in performance between the FA and the FAc algorithm;
- $H_d$: no difference in performance between the FA and the FAd algorithm.

**Table 2.4** End of run results for each algorithmic variant for F1–F6 (30D case)

| Algorithm | | Function 1 | Function 2 | Function 3 | Function 4 | Function 5 | Function 6 |
|---|---|---|---|---|---|---|---|
| FA | Best | 23.20 | 5,711,602 | 4.75 | 92.64 | 1269.10 | 8572.38 |
| | Mean | 31.60 | 10,997,353 | 5.18 | 110.91 | 1605.36 | 9109.52 |
| | Std | 4.09 | 1,809,538 | 0.15 | 9.76 | 123.22 | 217.68 |
| FAa | Best | 116.19 | 126,828,764 | 8.11 | 423.36 | 5755.66 | 7093.78 |
| | Mean | 160.83 | 229,847,840 | 9.03 | 550.35 | 7076.47 | 7563.07 |
| | Std | 17.70 | 38,217,754 | 0.29 | 57.36 | 606.75 | 263.16 |
| | $H_a$ | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| FAb | Best | 22.95 | 2,941,104 | 4.63 | 68.27 | 1141.99 | 8476.14 |
| | Mean | 28.31 | 7,045,931 | 5.06 | 96.18 | 1426.84 | 9228.50 |
| | Std | 2.15 | 1,817,708 | 0.15 | 9.30 | 105.46 | 193.53 |
| | $H_b$ | 0.0003 | 0.0000 | 0.0035 | 0.0000 | 0.0000 | 0.0291 |
| FAc | Best | 25.47 | 9,864,494 | 4.89 | 100.96 | 1400.92 | 8129.12 |
| | Mean | 33.90 | 15,997,277 | 5.19 | 119.82 | 1734.15 | 9224.45 |
| | Std | 3.66 | 2,195,339 | 0.13 | 9.56 | 135.65 | 345.81 |
| | $H_c$ | 0.0254 | 0.0000 | 0.8633 | 0.0007 | 0.0003 | 0.1289 |
| FAd | Best | 67.55 | 8,788,5 | 7.48 | 334.16 | 4275.20 | 6603.14 |
| | Mean | 125.18 | 130,952,522 | 8.31 | 432.63 | 5439.34 | 7291.36 |
| | Std | 27.59 | 63,633,167 | 0.40 | 65.45 | 781.83 | 238.38 |
| | $H_d$ | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| FA1 | Best | 5.87 | 769,851 | 3.24 | 30.26 | 605.85 | 9085.49 |
| | Mean | 11.95 | 2,016,472 | 3.82 | 40.37 | 804.36 | 9638.35 |
| | Std | 2.73 | 863,653 | 0.22 | 7.76 | 109.39 | 197.50 |
| | $H_1$ | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| FA2 | Best | 142.02 | 102,254,451 | 8.65 | 391.98 | 5136.39 | 6705.87 |
| | Mean | 161.94 | 219,256,368 | 9.03 | 537.83 | 6996.73 | 7498.80 |
| | Std | 11.31 | 42,204,038 | 0.19 | 56.26 | 553.66 | 320.17 |
| | $H_2$ | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| FA3 | Best | 54.24 | 34,303,799 | 6.69 | 174.07 | 2803.55 | 8402.80 |
| | Mean | 118.95 | 147,475,612 | 8.08 | 375.25 | 4866.90 | 9358.78 |
| | Std | 40.72 | 76,584,873 | 0.90 | 104.84 | 1748.87 | 403.90 |
| | $H_3$ | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0043 |

### 2.5.2 *Discussion of Results*

Initially we overview Figs. 2.6 and 2.7 to get an idea of the general trends in the results. Taking a high-level perspective, we note that while the performance of each algorithmic variant varies depending on the test function examined, the performance of the canonical version of FA is generally better than that of FA2 or FA3 (which

**Table 2.5** End of run results for each algorithmic variant for F7–F12 (30D case)

| Algorithm | | Function 7 | Function 8 | Function 9 | Function 10 | Function 11 | Function 12 |
|---|---|---|---|---|---|---|---|
| FA | Best | 43609.53 | 33,251,772,403 | –119.15 | 1961.38 | 341.18 | 69551.47 |
| | Mean | 73399.67 | 51,123,866,487 | –119.06 | 3363.09 | 429.12 | 153085.92 |
| | Std | 7333.86 | 5,270,435,371 | 0.03 | 426.19 | 35.67 | 45548.08 |
| FAa | Best | 60402.13 | 36,139,582,367 | –119.18 | 1687.58 | 273.99 | 73671.21 |
| | Mean | 73453.61 | 52,326,755,580 | –119.06 | 2343.38 | 345.95 | 92338.86 |
| | Std | 6757.77 | 7,917,980,119 | 0.03 | 241.66 | 35.97 | 11512.57 |
| | $H_a$ | 0.9765 | 0.4913 | 0.7135 | 0.0000 | 0.0000 | 0.0000 |
| FAb | Best | 48692.70 | 18,887,598,328 | –119.19 | 2137.20 | 246.97 | 68000.39 |
| | Mean | 56701.66 | 29,121,822,007 | –119.07 | 2684.83 | 307.88 | 123883.58 |
| | Std | 4172.39 | 3,406,108,447 | 0.05 | 196.01 | 29.30 | 46059.22 |
| | $H_b$ | 0.0000 | 0.0000 | 0.1841 | 0.0000 | 0.0000 | 0.0165 |
| FAc | Best | 63122.53 | 46,219,215,386 | –119.15 | 2833.47 | 399.57 | 69209.61 |
| | Mean | 84807.43 | 65,737,688,501 | –119.02 | 3928.29 | 518.35 | 162667.01 |
| | Std | 8166.06 | 11,155,657,744 | 0.05 | 462.02 | 58.75 | 50988.65 |
| | $H_c$ | 0.0000 | 0.0000 | 0.0025 | 0.0000 | 0.0000 | 0.4459 |
| FAd | Best | 54346.58 | 11,983,840,968 | –119.26 | 1372.58 | 162.86 | 46895.82 |
| | Mean | 65107.04 | 29,184,496,221 | –119.14 | 1788.40 | 286.61 | 75499.57 |
| | Std | 5908.49 | 7,029,267,934 | 0.05 | 228.13 | 38.78 | 9330.60 |
| | $H_d$ | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| FA1 | Best | 63849.53 | 24,242,817,446 | –119.15 | 2921.08 | 356.39 | 63231.72 |
| | Mean | 78639.07 | 45,195,413,457 | –119.06 | 3887.02 | 476.55 | 127709.92 |
| | Std | 7724.06 | 10,278,935,692 | 0.03 | 483.17 | 50.81 | 45586.37 |
| | $H_1$ | 0.0092 | 0.0067 | 0.8309 | 0.0000 | 0.0001 | 0.0352 |
| FA2 | Best | 51504.81 | 35,277,599,066 | –119.10 | 1802.17 | 249.84 | 85342.68 |
| | Mean | 71047.83 | 53,586,077,306 | –119.05 | 2295.75 | 356.74 | 99470.47 |
| | Std | 8502.51 | 10,895,945,623 | 0.03 | 277.71 | 44.55 | 7656.39 |
| | $H_2$ | 0.2560 | 0.2698 | 0.2514 | 0.0000 | 0.0000 | 0.0000 |
| FA3 | Best | 60643.40 | 23,910,071,808 | –119.15 | 2913.74 | 341.82 | 74311.54 |
| | Mean | 101787.20 | 83,404,345,263 | –118.97 | 4014.97 | 573.92 | 144463.81 |
| | Std | 16783.55 | 27,221,806,065 | 0.07 | 567.37 | 105.95 | 37769.59 |
| | $H_3$ | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.4281 |

have DP and IP turned off respectively), but that FA1 variant (in which momentum is turned off) appears to perform better than FA on several problems. Looking at the three variants FA1–FA3, FA1 performs better than either of the other two variants, with FA2 generally slightly outperforming FA3.

Taking the results together, it appears that DP (distributed perception) and IP (individual perception) contribute usefully to the search process but that the importance of momentum is not clearly demonstrated. It also appears that the DP and IP

**Table 2.6** End of run results for each algorithmic variant for F1–F6 (60D case)

| Algorithm | | Function 1 | Function 2 | Function 3 | Function 4 | Function 5 | Function 6 |
|---|---|---|---|---|---|---|---|
| FA | Best | 287.79 | 311,582,651 | 6.10 | 850.62 | 4602.89 | 17392.30 |
| | Mean | 379.79 | 607,913,058 | 9.21 | 1305.69 | 15203.91 | 18062.21 |
| | Std | 40.32 | 88,975,366 | 1.05 | 142.69 | 2687.78 | 324.02 |
| FAa | Best | 316.91 | 513,332,990 | 9.16 | 1230.04 | 14316.41 | 16740.36 |
| | Mean | 394.05 | 631,994,795 | 9.66 | 1386.62 | 16719.04 | 17920.98 |
| | Std | 32.12 | 64,533,215 | 0.22 | 81.55 | 1105.92 | 449.75 |
| | $H_a$ | 0.1355 | 0.2350 | 0.0250 | 0.0091 | 0.0060 | 0.1682 |
| FAb | Best | 77.71 | 26,742,064 | 5.49 | 243.21 | 3860.06 | 19609.69 |
| | Mean | 88.42 | 36,348,031 | 5.92 | 307.41 | 4223.62 | 20443.04 |
| | Std | 4.12 | 4,381,860 | 0.11 | 18.18 | 163.05 | 302.90 |
| | $H_b$ | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| FAc | Best | 85.60 | 43,270,190 | 5.62 | 305.97 | 3916.05 | 17972.25 |
| | Mean | 237.08 | 449,166,558 | 7.65 | 776.13 | 7813.33 | 18633.15 |
| | Std | 121.05 | 247,705,930 | 1.58 | 459.70 | 5026.22 | 387.10 |
| | $H_c$ | 0.0000 | 0.0016 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| FAd | Best | 312.57 | 371,304,375 | 8.89 | 1013.70 | 12571.32 | 16910.43 |
| | Mean | 388.55 | 589,343,313 | 9.70 | 1360.94 | 16819.07 | 17670.53 |
| | Std | 33.65 | 104,217,530 | 0.27 | 118.10 | 1412.29 | 322.42 |
| | $H_d$ | 0.2795 | 0.4060 | 0.0011 | 0.0544 | 0.0003 | 0.0000 |
| FA1 | Best | 25.57 | 2,635,463 | 3.83 | 82.36 | 1444.85 | 19840.50 |
| | Mean | 30.98 | 4,880,904 | 4.18 | 107.38 | 1847.15 | 21007.05 |
| | Std | 3.58 | 1,429,855 | 0.14 | 11.29 | 159.25 | 443.44 |
| | $H_1$ | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| FA2 | Best | 322.44 | 492,743,476 | 9.16 | 1017.49 | 15048.75 | 17191.05 |
| | Mean | 387.32 | 622,753,537 | 9.62 | 1334.03 | 16640.72 | 18069.47 |
| | Std | 24.19 | 69,381,378 | 0.18 | 109.47 | 816.48 | 392.15 |
| | $H_2$ | 0.3843 | 0.4742 | 0.0393 | 0.3916 | 0.0069 | 0.9380 |
| FA3 | Best | 322.44 | 492,743,476 | 8.39 | 961.99 | 12802.50 | 19901.43 |
| | Mean | 389.43 | 632,722,577 | 9.62 | 1303.44 | 16563.01 | 20945.41 |
| | Std | 26.05 | 73,067,155 | 0.33 | 137.22 | 1230.47 | 416.20 |
| | $H_3$ | 0.2764 | 0.2427 | 0.0472 | 0.9508 | 0.0146 | 0.0000 |

mechanisms can produce relatively similar levels of performance by the end of each experiment.

Next, we take a high-level overview of the performance of FA versus the algorithmic variants with different parameter settings (FAa–FAd). As before, the performance of the algorithmic variants depends on the test problem but in general, the ordering of performance (on the 60D cases in the Figures) appears to be $FAb > FAc > FA > FAd > FAa$. This ordering is plausible as fish in the algorithmic

**Table 2.7** End of run results for each algorithmic variant for F7–F12 (60D case)

| Algorithm | | Function 7 | Function 8 | Function 9 | Function 10 | Function 11 | Function 12 |
|---|---|---|---|---|---|---|---|
| FA | Best | 159458.20 | 101,440,099,772 | –118.92 | 5367.39 | 1163.32 | 330672.22 |
| | Mean | 210503.03 | 188,388,985,169 | –118.80 | 7107.06 | 1420.50 | 397832.08 |
| | Std | 19728.45 | 32,095,577,831 | 0.03 | 726.08 | 89.72 | 37187.04 |
| FAa | Best | 173831.05 | 142,801,200,876 | –118.92 | 5655.23 | 1180.41 | 284368.86 |
| | Mean | 213184.06 | 212,224,115,674 | –118.80 | 7197.15 | 1420.27 | 374998.36 |
| | Std | 17464.80 | 21,633,796,969 | 0.04 | 823.27 | 107.99 | 44435.06 |
| | $H_a$ | 0.5794 | 0.0013 | 0.6089 | 0.6547 | 0.9928 | 0.0350 |
| FAb | Best | 108596.74 | 59,065,713,240 | –118.88 | 3658.36 | 884.23 | 311357.75 |
| | Mean | 130188.82 | 70,454,662,207 | –118.80 | 5137.79 | 1079.93 | 713842.41 |
| | Std | 7705.40 | 6,690,493,140 | 0.03 | 686.85 | 64.09 | 338861.85 |
| | $H_b$ | 0.0000 | 0.0000 | 0.9847 | 0.0000 | 0.0000 | 0.0000 |
| FAc | Best | 150520.29 | 106,064,987,767 | –118.85 | 4919.17 | 1124.88 | 308881.61 |
| | Mean | 181714.55 | 166,416,739,386 | –118.79 | 7095.28 | 1330.02 | 521952.55 |
| | Std | 13605.42 | 36,340,990,005 | 0.03 | 1036.29 | 92.02 | 108045.25 |
| | $H_c$ | 0.0000 | 0.0160 | 0.3545 | 0.9595 | 0.0003 | 0.0000 |
| FAd | Best | 192437.89 | 125,451,253,553 | –118.93 | 5592.48 | 1231.55 | 239554.04 |
| | Mean | 218978.72 | 190,448,882,676 | –118.86 | 7100.64 | 1426.49 | 296261.74 |
| | Std | 12060.85 | 27,148,876,846 | 0.03 | 662.18 | 89.36 | 27396.78 |
| | $H_d$ | 0.0134 | 0.7504 | 0.0000 | 0.9666 | 0.7655 | 0.0000 |
| FA1 | Best | 146554.32 | 66,996,750,338 | –118.89 | 6427.89 | 1059.54 | 351171.76 |
| | Mean | 162332.97 | 86,987,862,848 | –118.80 | 7144.74 | 1162.22 | 864918.75 |
| | Std | 9607.24 | 11,310,775,223 | 0.03 | 404.89 | 57.04 | 346570.48 |
| | $H_1$ | 0.0000 | 0.0000 | 0.8134 | 0.8048 | 0.0000 | 0.0000 |
| FA2 | Best | 187090.19 | 140,251,557,412 | –118.85 | 5772.94 | 1132.50 | 301394.25 |
| | Mean | 211949.44 | 200,622,613,125 | –118.79 | 7009.69 | 1404.10 | 393418.16 |
| | Std | 11137.19 | 21,475,434,745 | 0.02 | 705.40 | 111.61 | 46386.99 |
| | $H_2$ | 0.7278 | 0.0880 | 0.6529 | 0.6003 | 0.5329 | 0.6858 |
| FA3 | Best | 197106.30 | 192,083,954,497 | –118.82 | 6664.71 | 1300.24 | 314263.05 |
| | Mean | 251792.25 | 274,617,244,480 | –118.69 | 9179.32 | 1707.49 | 664776.88 |
| | Std | 20690.26 | 44,967,767,801 | 0.03 | 1189.09 | 157.68 | 193674.39 |
| | $H_3$ | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |

variant FAb have a wider ‘perception radius’ than do the fish in any of the other algorithmic variants allowing them to perceive information from a greater volume of the search space. Conversely, the relatively poorer search performance of FAa is not unexpected as it has a smaller perception radius than the other algorithm variants.

Hence, from a high-level overview of Figs. 2.6 and 2.7, the key points are that while IP and DP provide useful information for the search process, the momentum mechanism does not appear to be as important. It is also evident that the performance

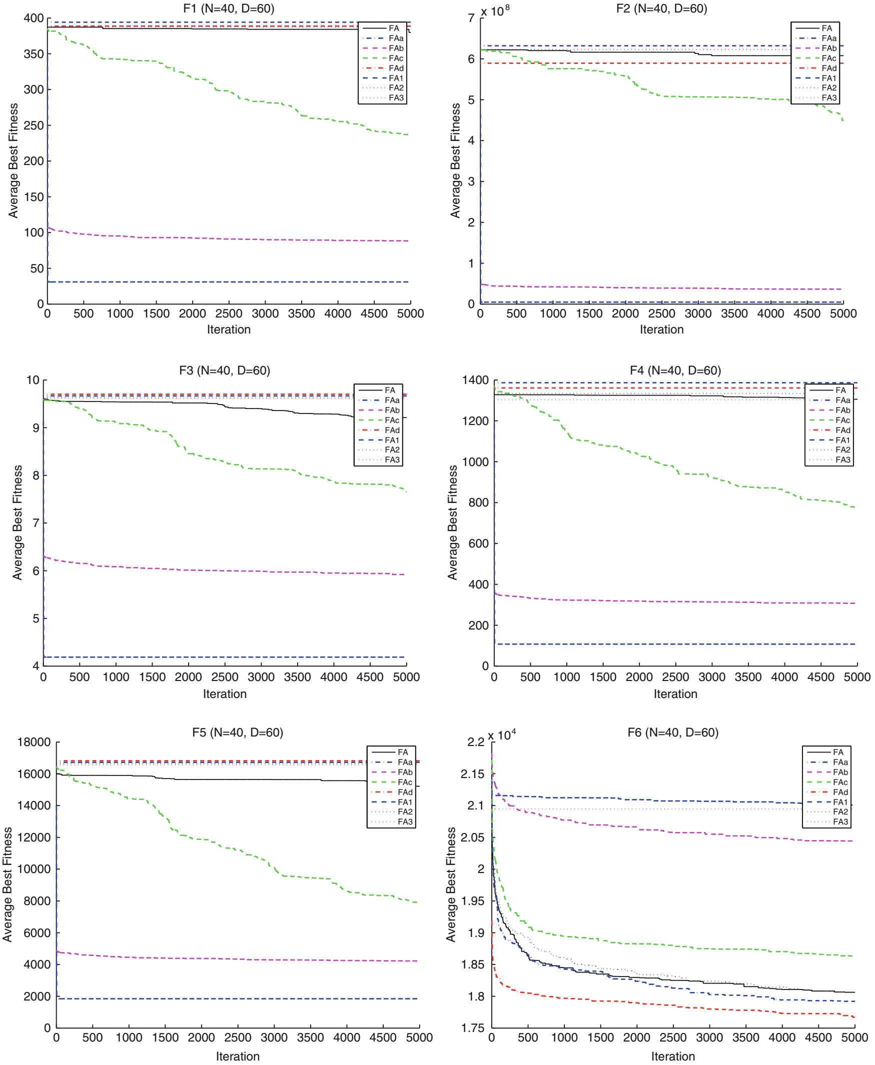


**Fig. 2.6** Average best performance (averaged over 30 trials) of each algorithm variant on test problems F1–F6 (60D)

of the algorithm is sensitive to choices of perception radius, with increases in this parameter leading to enhanced performance.

Next, we proceed to look at the results in Tables 2.4, 2.5, 2.6 and 2.7 in order to obtain finer detail.

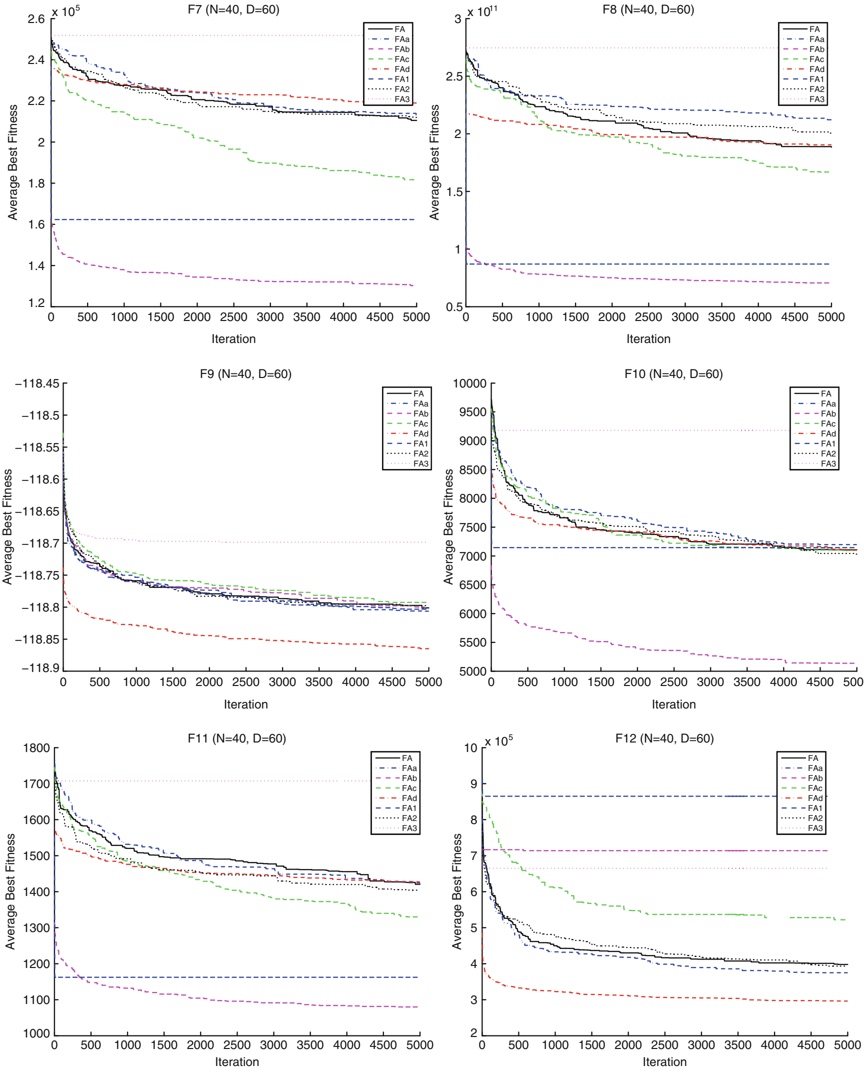


**Fig. 2.7** Average best performance (averaged over 30 trials) of each algorithm variant on test problems F7–F12 (60D)

### 2.5.3 Analysis of Components in FA

Comparing the mean (of the best results found across each of the 30 trials) performance of FA with FA1, FA outperforms FA1 in 5 out of 12 cases (30D) and 4 out of 12 cases (60D). In all but one case, the difference in mean performance between the algorithms is significant. Hence, the conclusion drawn is that there is no compelling evidence that the addition of a momentum mechanism has led to enhanced search performance. It is noted that the inclusion of a momentum mechanism in these experiments was motivated by general findings in the behavioural ecology literature [39] that organisms tend to display directional persistence rather than it being a distinct mechanism displayed by golden shiner fish [3].

Comparing the performance of FA with FA2, we note that FA outperforms FA2 in 7 out of 12 cases (30D) and 8 out of 12 cases (60D). In 9 cases (30D) and 1 case (60D) the difference is statistically significant. The conclusion drawn is that FA slightly outperforms FA2, but that the degree of outperformance becomes less (statistically speaking) as we move to the 60D case.

Comparing FA with FA3, FA outperforms FA3 in 11 out of 12 cases (30D) and 10 out of 12 cases (60D). In 11 cases (30D) and 8 cases (60D) the difference is statistically significant. The conclusion drawn is that FA generally outperforms FA3 and that, based on the results for FA2 and FA3, the inclusion of both IP and DP mechanisms (as distinct from only including one mechanism) produces a better quality search process.

We also compare the performance of FA2 and FA3, and find that FA2 outperforms FA3 in 7 out of 12 cases (30D) and 9 out of 12 cases (60D), indicating that a IP mechanism produces a better search performance than DP alone. This is not surprising as the DP mechanism is not driven by any feedback from the environment, and therefore, on its own is similar to a random search process. As would be expected, the standard deviation of the results produced by FA3 is generally higher than those produced by either FA1 or FA2.

Hence, the results suggest that while social information propagation can usefully spread information on good locations amongst the population of agents, it needs to be informed by information from the individual perception mechanism in order to strongly guide the search process. Combining the results, across the algorithmic variants we get a general performance ordering of $FA1 > FA > FA2 > FA3$.

### *2.5.4 Parameter Sensitivity Analysis*

The detailed 'end of run' results from the FAa, FAb FAc and FAd variant algorithms are shown in Tables 2.4, 2.5, 2.6 and 2.7. Initially, we compare the results of each algorithmic variant with the performance of the canonical algorithm FA.

We note that FA outperforms FAa in 8 out of 12 cases (30D) and 9 out of 12 cases (60D). In 9 cases (30D) and 4 cases (60D) these differences are statistically significant. This suggests that FA generally performs better than FAa, which is not unexpected given that FA has a wider perception radius.

Examining FA versus FAb, FA performs better in only 2 out of 12 cases (30D) and 1 out of 12 cases (60D). The differences in mean performance are statistically significant in 10 (30D) and 11 (60D) cases respectively. The strong performance of FAb arises as in this variant of the algorithm, the fish have a wider perception radius than they do in FA.

Comparing the results of FAa and FAb we note that FAa performs better on the majority of test problems. Combining the results from the above analyses, we can conclude that the choice of perception radius is a critical parameter for the algorithm.

The FAc variant employs 10 samples in each IP step. The canonical FA outperforms FAc in 12 out of 12 cases (30D) but in only 3 out of 12 cases (60D), with the differences in performance being significant in 8 (30D) and 10 (60D) cases respectively. It it interesting to note the switch in relative performance when the dimensionality of the test problems is increased.

In contrast to FAc, the variant FAd only undertakes a single sampling in each IP step. Comparing FA with FAd, FA performs better in 5 out of 12 cases (30D) and in 7 out of 12 cases (60D). The differences in performances are significant in 12 out of 12 cases (30D) and 6 out of 12 cases (60D).

Comparing FAc and FAd, it is not clearly evident that either outperforms the other, as the performance ranking between the two varies across the test problems. The conclusion is that the results from the FA algorithm are not clearly impacted by choice of number of IP samplings.

## 2.6 Conclusions

The propagation of information about the environment amongst a population via social communication has attracted increasing research interest in recent decades with the realisation that many animal species engage in subtle forms of information transfer which had previously escaped notice. More generally, we can consider this process as information flowing in a network of nodes or agents, wherein each agent receives inputs from their senses and from conspecifics, processes this information, and in turn through their resulting actions, subsequently influence actions of other agents.

In this study we draw inspiration from the schooling behaviour of ‘golden shiner’ fish which alter their movement in an effort to track shade and develop a novel optimization algorithm, the fish algorithm (FA). The FA can be considered as a swarm algorithm as the search process embeds bottom-up learning via information flow between agents (fish). We assess the utility of the algorithm on a series of test problems and undertake an analysis of the algorithm by examining the importance of its component elements for the search process. The results indicate that momentum or ‘directional persistence’ mechanism is not found to be particularly useful but that best results are obtained when using a mix of information from individual perception and social communication. While social communication can usefully spread information on good locations amongst the population of agents, it needs to be supplemented by information from the individual perception mechanism in order to strongly guide the search process.

The current study indicates several interesting areas for follow up research. Obviously the results from any study only extend to the test problems and specific parameter settings examined, and future work could seek to examine the utility of the algorithm in additional problem domains. A factor which is not fully included in current work is that fish do not select shoal mates randomly but rather prefer to shoal with healthy fish, and fish which are similar in size and age to themselves. The algorithms developed in this chapter could be adapted to incorporate these issues more comprehensively.

At an even deeper level, the results of the study highlight the question as to what is the optimal balance between the use of individual perception and the propagation of social information in the population? In other words, what weight should be placed on each factor in order to optimise the search process. Further investigation of this issue has potential to assist in our understanding as to how best to tailor optimization algorithms for specific problem environments, and for deepening our understanding of the foraging strategies of various organisms.

**Acknowledgments** This publication has emanated from research conducted with the financial support of Science Foundation Ireland under Grant Number 08/SRC/FM1389.

## References

1. Amintoosi, M., Fathy, M., Mozayani, N., Rahmani, A.: A fish school clustering algorithm: applied to student sectioning problem. In: Proceedings of 2007 International Conference on Life System Modelling and Simulation (LSMS), Published as a Supplementary Volume to Dynamics of Continuous Discrete and Impulse Systems, Series B: Applications and Algorithms, vol. 2, pp. 696–699. Watam Press, Canada (2007)
2. Bastos Filho, C., de Lima Neto, F., Lins, A., Nascimento, A., Lima, M.: A novel search algorithm based on fish school behavior. In: Proceedings of IEEE International Conference on Systems, Man and Cybernetics (SMC), pp. 2646–2651. IEEE Press, New York (2008)
3. Berdahl, A., Torney, C., Ioannou, C., Faria, J., Couzin, I.: Emergent sensing of complex environments by mobile animal groups. Science **339**, 574–576 (2013)
4. Bin, Z., Jianlin, M., Haiping, L.: A hybrid algorithm for sensing coverage problem in wireless sensor networks. In: Proceedings of IEEE International Conference on Cyber Technology in Automation, Control and Intelligent Systems, pp. 162–165. IEEE Press, Kunming (2011)
5. Bing, Z., Wen, D.: Scheduling arrival aircrafts on multi-runway based on improved artificial fish swarm algorithm. In: Proceedings of the 10th International Conference on Computational and Information Sciences (ICCIS '10), pp. 499–502. IEEE Press, New York (2010)
6. Bradbury, J., Vehrencamp, S.: Principles of Animal Communication, 2nd edn. Sinauer Associates, Sunderland, MA, USA (2011)
7. Bonabeau, E., Dorigo, M., Theraulaz, G.: Swarm Intelligence: From Natural to Artificial Systems. Oxford University Press, Oxford (1999)
8. Chong, C., Low, M., Sivakumar, A., Gay, K.: A bee colony optimization algorithm to job shop scheduling. In: Proceedings of the 2006 Winter Simulation Conference (WinterSim), pp. 1954–1961. IEEE Press, New Jersey (2006)
9. Dorigo, M.: Optimization, learning and natural algorithms. Ph.D. thesis, Politecnico di Milano (1992)
10. Dorigo, M., DiCaro, G.: Ant colony optimization: a new meta-heuristic. In: Proceedings of IEEE Congress on Evolutionary Computation (CEC), pp. 1470–1477. IEEE Press, Piscataway, NJ (1999)
11. Dorigo, M., Maniezzo, V., Colorni, A.: Ant system: optimization by a colony of cooperating agents. IEEE Trans. Syst. Man Cybern. **26**, 29–41 (1996)

12. Dorigo, M., Stützle, T.: Ant Colony Optimization. MIT Press, Cambridge (2004)
13. Engelbrecht, A.: Fundamentals of Computational Swarm Intelligence. Wiley, Chichester (2005)
14. Farzi, S.: Efficient job scheduling in grid computing with modified artificial fish swarm algorithm. Int. J. Comput. Theory Eng. **1**(1), 13–18 (2009)
15. Grunbaum, D., Viscido, S., Parrish, J.: Extracting interative control algorithms from group dynamics of schooling fish. In: Coop Control Lecture Notes in Control and Information Sciences (LNCIS 309), pp. 103–117. Springer (2004)
16. He, D., Qu, L., Guo, X.: Artificial fish-school algorithm for integer programming. In: Proceedings of IEEE International Conference on Information Engineering and Computer Science (ICIECS), pp. 1–4. IEEE Press, New York (2009)
17. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of the IEEE International Conference on Neural Networks, pp. 1942–1948. IEEE Press, Piscataway, NJ (1995)
18. Kennedy, J., Eberhart, R., Shi, Y.: Swarm Intelligence. Morgan Kaufman, San Mateo (2001)
19. Li, X., Shao, Z., Qian, J.: An optimizing method based on autonomous animats: fish swarm algorithm. Syst. Eng. Theory Pract. **22**, 32–38 (2002) (in Chinese)
20. Liang, J.J., Suganthan, P.N., Deb, K.: Novel composition test functions for numerical global optimization. In: Proceedings of IEEE Swarm Intelligence Symposium, pp. 68–75. IEEE Press, Chicago (2005)
21. Liu, B.-Q., Sun G.-C.: Artificial fish swarm algorithm for traffic network design problem. Comput. Eng. **37**(8), 161–163 (2011) (in Chinese)
22. Kavanau, J.: Vertebrates that never sleep: implications for sleep's basic function. Brain Res. Bull. **46**(4), 269–279 (1998)
23. Nakrani, S., Tovey, C.: On honey bees and dynamic server allocation in internet hosting centres. Adapt. Behav. **12**, 223–240 (2004)
24. Neshat, M., Sepidnam, G., Sargolzaei, M., Najaran Toosi, A.: Artificial fish swarm algorithm: a survey of the state-of-the-art, hybridization, combinatorial and indicative applications. Artif. Intell. Rev. (2012). doi:10.1007/s10462-012-9342-2. Accessed 6 May 2012
25. Meng, F., Zhao, H., Zhao, Q., Ma, W., Cao, Y., Wang, L.: Artificial fish swarm-based energy efficient qos classification algorithm to next generation electric power communication networks. Appl. Mech. Mater. **392**, 857–861 (2013)
26. Miller, N., Garnier, S., Hartnett, A., Couzin, I.: Both information and social cohesion determine collective decisions in animal groups. PNAS, **110**(13), 5263–5268 (2013)
27. Parrish, J., Viscido, S., Grunbaum, D.: Self-organized fish schools: an examination of emergent properties. Biol. Bull. **202**, 296–305 (2002)
28. Passino, K.: Distributed optimization and control using only a germ of intelligence. In: Proceedings of the 2000 IEEE International Symposium on Intelligent Control, pp. 5–13. IEEE Press, Patras, Greece (2000)
29. Passino, K.: Biomimicry of bacterial foraging for distributed optimization and control. IEEE Control Syst. Mag. **22**, 52–67 (2002)
30. Pham, D., Ghanbarzadeh, A., Koc, E., Otri, S., Rahim, S., Zaidi, M.: The bees algorithm—a novel tool for complex optimization problems. In: Proceedings of International Production Machines and Systems (IPROMS), pp. 454–459. Elsevier, Oxford (2006)
31. Reynolds, C.: Flocks, Herds and Schools, a distributed behavioral model. In: Proceedings of the 14th annual conference on computer graphics and interactive techniques (SIGGRAPH), pp. 25–34. Anaheim, California (1987)
32. Slobodchikoff, C., Perla, B., Verdolin, J.: Prairie Dogs: Communication and Community in an Animal Society. Harvard University Press, Cambridge, Massachusetts (2009)
33. Stocker, S.: Models for tuna formation. Math. Biosci. **156**, 167–190 (1999)
34. Suganthan, P.N., Hansen, N., Liang, J.J., Deb, K., Chen, Y.P., Auger, A., Tiwari, S.: Problem definitions and evaluation criteria for the CEC 2005 special session on Real-Parameter optimization. Nanyang Technological University, Technical report (2005)
35. Sumpter, D., Krause, J., James, R., Couzin, I., Ward, A.: Consensus decision making by fish. Curr. Biol. **18**, 1773–1777 (2008)

36. Tian, W., Liu, J.: An improved artificial fish swarm algorithm for multi robot task scheduling. In: Proceedings of the 2009 IEEE 5th International Conference on Natural Computation, pp. 127–130. IEEE Press, New York (2009)
37. Tian, W., Tian, Y., Ai, L., Liu, J.: A new optimization algorithm for fuzzy set design. In: Proceedings of the 2009 IEEE International Conference on Intelligent Human-Machine Systems and Cybernetics, pp. 431–435. IEEE Press, New York (2009)
38. Tsai, H.-C., Lin, Y.-H.: Modification of the fish swarm algorithm with particle swarm optimization formulation and communication behavior. Appl. Soft Comput. **11**, 5367–5374 (2011)
39. Viswanathan, G., da Luz, M., Raposo, E., Stanley, E.: The Physics of Foraging: An Introduction to Random Searches and Biological Encounters. Cambridge University Press, Cambridge (2011)
40. Wang, C.-R., Zhou, C.-L., Ma, J.-W.: An improved artificial fish-swarm algorithm and its application in feed-forward neural networks. In: Proceedings of the 2005 IEEE International Conference on Machine Learning and Cybernetics, vol. 5, pp. 2890–2894. IEEE Press (2005)
41. Yang, X.-S.: Engineering optimization via nature-inspired virtual bee algorithms. In: Mira, J., Álvarez, J. (eds.) Artificial Intelligence and Knowledge Engineering Applications: A Bioinspired Approach, First International Work-Conference on the Interplay Between Natural and Artificial Computation (IWINAC 2005), pp. 317–323. Springer, Berlin (2005)
42. Yiyue, W., Hongmei, L., Hengyang, H.: Wireless sensor network deployment using an optimized artificial fish swarm algorithm. In: Proceedings of 2012 IEEE International Conference on Computer Science and Electronics Engineering, vol. 2, pp. 90–94. IEEE Press, Hangzhou (2012)
43. Zhang, K., Zhang, W., Dai, C.-Y., Zeng, J.-Z.: Artificial fish-swarm based coverage-enhancing algorithm for visible light sensor networks. Optoelectron. Lett. **6**(3), 229–231 (2010)
44. Zhou, Y., Liu, B.: Two novel swarm intelligence clustering analysis methods. In: Proceedings of the 5th International Conference on Natural Computation, pp. 497–501. IEEE Press, Tianjin (2009)

# Chapter 3
# Models for Trust Inference in Social Networks

**Cai-Nicolas Ziegler and Jennifer Golbeck**

**Abstract** Interpersonal trust between any two people in social networks is hard to gauge, and even harder to *infer*, given that these two people are not connected by an immediate social link, such as friendship or acquaintanceship. In order to be able to make accurate inferences for an arbitrary tuple of people in a given social environment, we present an approach, named Appleseed, that is based on mechanics taken from neuropsychology, known as spreading activation models. Compelling in its simplicity, we relate the concept to trust propagation and evaluation in an intuitive fashion. While Appleseed works very well when paths between two arbitrary people in the network can be established, no inference of trust is possible when this is not the case. To this end, we present several algorithms for inferring trust that go beyond network structure and demonstrate their accuracy in real social networks. We also show how these algorithms can be augmented with additional data that may be available in some contexts.

## 3.1 Introduction

In our world of information overload and global connectivity leveraged through the Web and other media types, social trust [29] between individuals becomes an invaluable and precious good. Hereby, trust exerts an enormous impact on decisions whether to believe or disbelieve information asserted by other peers. Belief should only be accorded to statements from people we deem trustworthy. However, when supposing huge social networks such as the case for social media platforms, trust judgements based on personal experience and acquaintanceship become unfeasible.

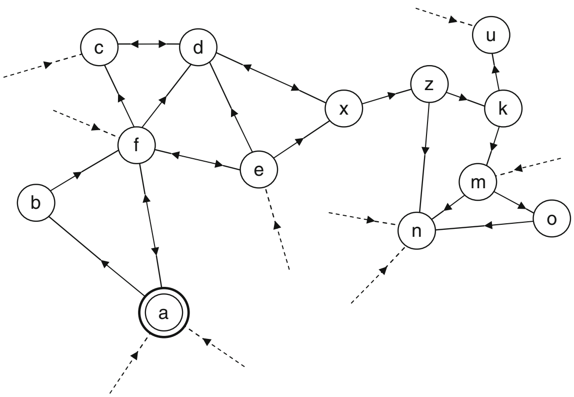
The research presented in this chapter has been conducted while being affiliated with Albert-Ludwigs- Universität Freiburg i.Br., Germany.

C.-N. Ziegler (✉)
XING EVENTS GmbH, Sandstraße 33, 80335 Munich, Germany
e-mail: cai-nicolas.ziegler@xing.com

J. Golbeck
University of Maryland, College Park, MD 20742, USA
e-mail: jgolbeck@umd.edu

D. Król et al. (eds.), *Propagation Phenomena in Real World Networks*,
Intelligent Systems Reference Library 85, DOI 10.1007/978-3-319-15916-4_3

**Fig. 3.1** Sample web of trust for agent *a*



In general, we accord trust, which has been defined as the "subjective expectation an agent has about another's future behavior based on the history of their encounters" [30], to only small numbers of people. These people, again, trust another limited set of people, and so forth. The network structure emanating from our person (see Fig. 3.1), composed of trust statements linking individuals, constitutes the basis for trusting people we do not know personally.

We might be tempted to adopt the policy of trusting all those people who are trusted by persons we trust. Trust would thus propagate through the network [21] and become accorded whenever two individuals can reach each other via at least one trust path. However, common sense tells us we should not rely upon this strategy. More complex metrics are needed in order to more sensibly evaluate trust between two persons. Among other features, these trust metrics must take into account social and psychological aspects of trust and suffice criteria of computability and scalability likewise.

When adopting the most basic policy of trust propagation, all those people who are trusted by persons we trust are considered likewise trustworthy. Trust would thus propagate through the network and become accorded whenever two individuals can reach each other via at least one trust path. However, owing to certain implications of interpersonal trust, e.g., attack-resistance, trust decay, etc., more complex metrics are needed to sensibly evaluate social trust. Subtle social and psychological aspects must be taken into account and specific criteria of computability and scalability satisfied.

In this chapter, we aim at designing one such complex trust metric,[1] particularly tailored to social filtering tasks by virtue of its ability to infer continuous trust values through fixpoint iteration, rendering ordered trust-rank lists feasible.

However, one challenge to using network data for trust inference is that when there are few or no paths to a node in the network, the algorithms may not be able

[1] Note that trust concepts commonly adopted for webs of trust, and similar trust network applications, are largely general and do not cover specifics such as "situational trust" [26], as has been pointed out in [13]. For instance, agent $a_i$ may blindly trust $a_j$ with respect to books, but not trust $a_j$ with respect to trusting others, for $a_j$ has been found to accord trust to other people too easily. For our trust propagation scheme at hand, we also suppose this largely uni-dimensional concept of trust.

to infer a value. Using TidalTrust, another network-based trust inference algorithm, and data from FilmTrust, a movie rating website with an underlying trust network, we show that integrating additional, non-network-based information into the trust computation, can improve the number of node pairs for which a trust value can be calculated. Integrating the network and data-based models can also improve the accuracy of these algorithms.

### *3.1.1 Trust Representation and Model*

We assume that all trust information is publicly accessible for any agent in the system, e.g., through machine-readable personal homepages distributed over the network. Agents $a_i \in A = \{a_1, a_2, \ldots, a_n\}$ are associated with a partial trust function $W_i \in T = \{W_1, W_2, \ldots, W_n\}$ each, where $W_i : A \to [0, 1]^{\perp}$ holds, which corresponds to the set of trust assertions that $a_i$ has stated.

In most cases, functions $W_i(a_j)$ will be very sparse as the number of individuals an agent is able to assign explicit trust ratings for is much smaller than the total number $n$ of agents. Moreover, the higher the value of $W_i(a_j)$, the more trustworthy $a_i$ deems $a_j$. Conversely, $W_i(a_j) = 0$ means that $a_i$ considers $a_j$ to be *not trustworthy*. The assignment of trust through continuous values between 0 and 1, and their adopted semantics, is in perfect accordance with [26], where possible stratifications of trust values are proposed. Our trust model defines one directed trust graph with nodes being represented by agents $a_i \in A$, and directed edges from nodes $a_i$ to nodes $a_j$ representing trust statements $W_i(a_j)$.

For convenience, we introduce the partial function $W : A \times A \to [0, 1]^{\perp}$, which we define as the union of all partial functions $W_i \in T$.

### *3.1.2 Overview of Trust Metrics for Social Networks*

Trust and reputation ranking metrics have primarily been used for the Public Key Infrastructure (PKI) [4, 23, 28, 33, 34], rating and reputation systems part of online communities [14, 22, 24], peer-to-peer networks [3, 18–20, 36], and also mobile computing [7]. Each of these scenarios favors different trust metrics. For instance, reputation systems for online communities tend to make use of *centralized trust servers* that compute global trust values for all users on the system [14]. On the other hand, peer-to-peer networks of moderate size rely upon distributed approaches that are in most cases based upon PageRank [18, 36].

Larger social networks, such as the Semantic Web, are made up of millions of nodes, i.e., agents. The fitness of *distributed* approaches to trust metric computation, such as described in [18, 35], hence becomes limited for various reasons:

*Trust data storage.* Every agent $a_i$ needs to store trust rating information about any other agent $a_j$ on the network. Agent $a_i$ uses this information in order to merge it with own trust beliefs and propagates the synthesized information to his trusted agents [22]. For a network organized in a decentralized fashion, the number of agents for whom to keep trust information will still exceed the storage capacities of most nodes.

*Convergence.*The structure of networks not under centralized control is diffuse and commonly not subject to some higher ordering principle or hierarchy. Furthermore, the process of trust propagation is *necessarily asynchronous* as there is no central node of authority. Convergence of trust values might thus take a very long time.

The huge advantage of distributed approaches to trust propagation and computation, on the other hand, is the *immediate availability* of computed trust information about any other agent $a_j$ in the system. Moreover, agents have to disclose their trust assertions only to peers they actually *trust* [35]. For instance, suppose that $a_i$ declares his trust in $a_j$ by $W_i(a_j) = 0.1$, which is very low. Hence, $a_i$ might want $a_j$ not to know about that fact. As distributed metrics only propagate *synthesized* trust values from nodes to successor nodes in the trust graph, $a_i$ would not have to openly disclose his trust statements to $a_j$.

As it comes to centralized, i.e., *locally computed*, metrics, *full* trust information access is required for agents inferring trust. Hence, online communities based on trust require their users to disclose all trust information to the community server, but not necessarily to other peers [14]. Privacy thus remains preserved. On social networks such as the Semantic Web, however, there is no such central authority that computes trust. *Any* agent might want to do so. Our own trust model, as well as trust models proposed in [1, 7, 13], are hence based upon the assumption of *publicly available trust information*. Though privacy concerns may persist, this assumption is vital, owing to the afore-mentioned deficiencies of distributed computation models. Moreover, centralized *global* metrics, such as depicted in [14, 31], also fail to fit our requirements: because of the huge number of agents issuing trust statements, only dedicated server clusters could be able to manage the whole bulk of trust relationships.

Scalar metrics, e.g., PKI proposals [4, 23, 28, 33, 34] and those metrics described in [13], have poor scalability properties, owing to exponential time complexity [33].

Consequently, we advocate *local group* trust metrics [43] for the Semantic Web and other large-scale decentralized networks. Local group trust metrics do not only compute trust values for a specified pair of agents, $(a_i, a_j) \in V \times V$, but compute trust *ranks* for *sets* of individuals from $V$. The predicate *local* refers to the metric's network perspective, which is subjective, adopting the position of one of the agents. That is, trust values assigned to $a_j \in V$ are different for two different trust sources $a_i$.

Local group trust metrics bear several welcome properties with respect to computability and complexity, which may be summarized as follows:

*Partial trust graph exploration.* Global metrics require a priori full knowledge of the entire trust network. Distributed metrics store trust values for all agents in the system, thus implying massive data storage demands. On the other hand, when computing trusted *neighborhoods*, the trust network only needs to be explored partially: originating from the trust source, one only follows those trust edges that seem promising, i.e., bearing high trust weights, and which are not too far away from the trust source. Inspection of agent nodes is thus performed in a just-in-time fashion. Hence, prefetching bulk trust information is not required.

*Computational scalability.* Tightly intertwined with partial trust graph exploration is computational complexity. Local group trust metrics scale well to any social network size, as only tiny subsets of relatively constant size[2] are visited.

## 3.2 Design of Local Group Trust Metrics

Local group trust metrics, in their function as means to compute trust neighborhoods, have not been subject to mainstream research so far. Significant research has effectively been limited to the work done by Levien [22] who has conceived and deployed the Advogato group trust metric. This section provides an overview of Advogato and introduces our own Appleseed trust metric, eventually comparing both approaches.

### *3.2.1 Outline of Advogato Maxflow*

The Advogato maximum flow trust metric has been proposed by Levien and Aiken [24] in order to discover which users are trusted by members of an online community and which are not. Trust is computed through one centralized community server and considered relative to a seed of users enjoying supreme trust. However, the metric is not only applicable to community servers, but also to *arbitrary* agents which may compute *personalized lists* of trusted peers, not only one single global ranking for the whole community they belong to. In this case, the active agent himself constitutes the singleton trust seed. The following paragraphs briefly introduce Advogato's basic concepts. For more detailed information, refer to [22–24].

#### 3.2.1.1 Trust Computation Steps

Local group trust metrics compute sets of agents trusted by those being part of the trust seed. In case of Advogato, its input is given by an integer number $n$, which is supposed to be equal to the number of members to trust [24], as well as the trust seed $s$, which is a subset of the entire set of users $A$. The output is a *characteristic function* that maps each member to a boolean value indicating his trustworthiness:

$$\text{Trust}_M : 2^A \times \mathbb{N}_0^+ \rightarrow (A \rightarrow \{\text{true}, \text{false}\}) \tag{3.1}$$

The trust model underlying Advogato does *not* provide support for weighted trust relationships in its original version.[3] Hence, trust edges extending from individual $x$

[2] Supposing identical parameterizations for the metrics in use, as well as similar network structures.

[3] Though various levels of peer certification exist, their interpretation does not perfectly align with weighted trust relationships.

to $y$ express *blind*, i.e., *full*, trust of $x$ in $y$. The metrics for PKI maintenance suppose similar models. Maximum integer network flow computation [8] has been investigated by Reiter and Stubblebine [33, 34] in order to make trust metrics more reliable. Levien adopted and extended this approach for group trust in his Advogato metric.

Capacities $C_A : A \rightarrow \mathbb{N}$ are assigned to every community member $x \in A$ based upon the shortest-path distance from the seed to $x$. Hereby, the capacity of the seed itself is given by the input parameter $n$ mentioned before, whereas the capacity of each successive distance level is equal to the capacity of the previous level $l$ divided by the average outdegree of trust edges $e \in E$ extending from $l$. The trust graph we obtain hence contains one single source, which is the set of seed nodes considered as one single "virtual" node, and multiple sinks, i.e., all nodes other than those defining the seed. Capacities $C_A(x)$ constrain nodes. In order to apply Ford-Fulkerson maximum integer network flow [8], the underlying problem has to be formulated as single-source/single-sink, having capacities $C_E : E \rightarrow \mathbb{N}$ constrain *edges* instead of *nodes*. Hence, Algorithm 3.1 is applied to the old directed graph $G = (A, E, C_A)$, resulting in a new graph structure $G' = (A', E', C_{E'})$ (Fig. 3.2).

Figure 3.3 depicts the outcome of converting node-constrained single-source/multiple-sink graphs (see Fig. 3.2) into single-source/single-sink ones with capacities constraining edges.

Conversion is followed by simple integer maximum network flow computation from the trust seed to the super-sink. Eventually, the trusted agents $x$ are exactly those peers for whom there is flow from "negative" nodes $x^-$ to the super-sink. An additional constraint needs to be introduced, requiring flow from $x^-$ to the super-sink whenever there is flow from $x^-$ to $x^+$. The latter constraint assures that node $x$ does

```
func transform ($G = (A, E, C_A)$) {
  set $E' \leftarrow \emptyset$, $A' \leftarrow \emptyset$;
  for all $x \in A$ do
    add node $x^+$ to $A'$;
    add node $x^-$ to $A'$;
    if $C_A(x) \geq 1$ then
      add edge $(x^-, x^+)$ to $E'$;
      set $C_{E'}(x^-, x^+) \leftarrow C_A(x) - 1$;
      for all $(x, y) \in E$ do
        add edge $(x^+, y^-)$ to $E'$;
        set $C_{E'}(x^+, y^-) \leftarrow \infty$;
      end do
      add edge $(x^-, \text{supersink})$ to $E'$;
      set $C_{E'}(x^-, \text{supersink}) \leftarrow 1$;
    end if
  end do
  return $G' = (A', E', C_{E'})$;
}
```

**Algorithm 3.1**. Trust graph conversion

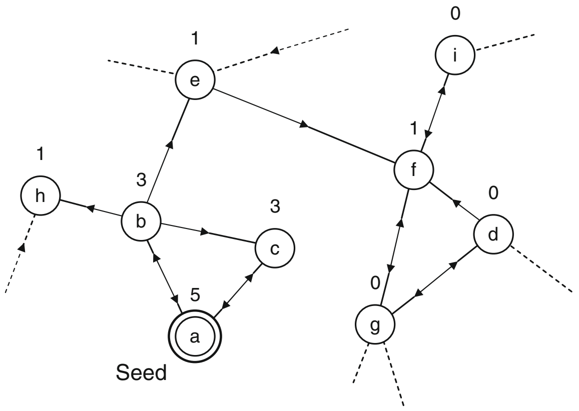


**Fig. 3.2** Trust graph *before* conversion for Advogato

not only serve as an intermediate for the flow to pass through, but is *actually added* to the list of trusted agents when reached by network flow. However, the standard implementation of Ford-Fulkerson traces shortest paths to the sink first [8]. The above constraint is thus satisfied implicitly already.

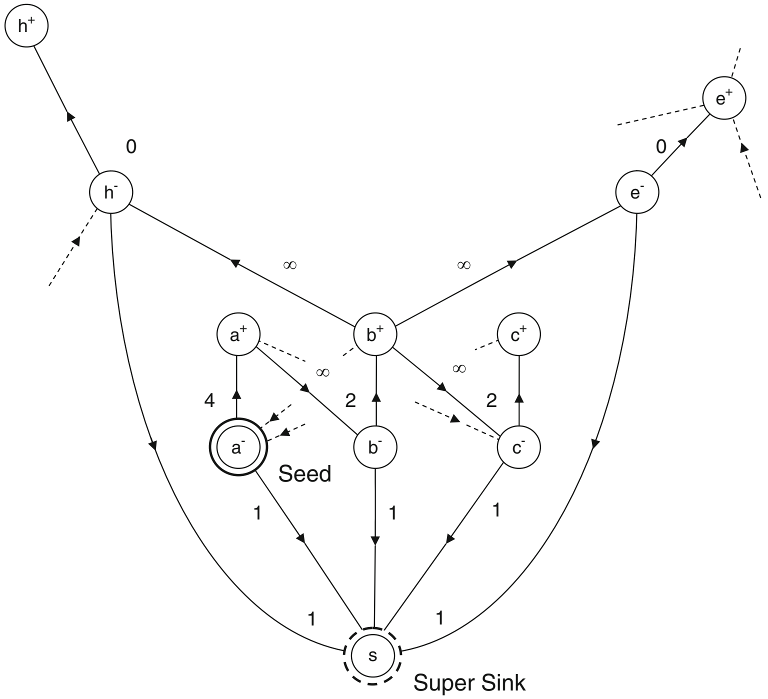


**Fig. 3.3** Trust graph *after* conversion for Advogato

*Example 3.1* (Advogato trust computation) Suppose the trust graph depicted in Fig. 3.2. The only seed node is $a$ with initial capacity $C_A(a) = 5$. Hence, taking into account the outdegree of $a$, nodes at unit distance from the seed, i.e., nodes $b$ and $c$, are assigned capacities $C_A(b) = 3$ and $C_A(c) = 3$, respectively. The average outdegree of both nodes is 2.5 so that second level nodes $e$ and $h$ obtain unit capacity. When computing maximum integer network flow, agent $a$ will accept himself, $b$, $c$, $e$, and $h$ as trustworthy peers.

#### 3.2.1.2 Attack-Resistance Properties

Advogato has been designed with resistance against massive attacks from malicious agents outside of the community in mind. Therefore, an upper bound for the number of "bad" peers chosen by the metric is provided in [24], along with an informal security proof to underpin its fitness. Resistance against malevolent users trying to break into the community can already be observed in the example depicted by Fig. 3.1, supposing node $n$ to be "bad": though agent $n$ is trusted by numerous persons, he is deemed less trustworthy than, for instance, $x$. While there are fewer agents trusting $x$, these agents enjoy higher trust reputation[4] than the numerous persons trusting $n$. Hence, it is not just the *number* of agents trusting an individual $i$, but also the trust *reputation* of these agents that exerts an impact on the trust assigned to $i$. PageRank [31] works in a similar fashion and has been claimed to possess properties of attack-resistance similar to those of the Advogato trust metric [22]. In order to make the concept of attack-resistance more tangible, Levien proposes the "bottleneck property" as a common feature of attack-resistant trust metrics. Informally, this property states that the "trust quantity accorded to an edge $s \to t$ is not significantly affected by changes to the successors of $t$" [22].

Attack-resistance features of various trust metrics are discussed in detail in [23, 38].

### *3.2.2 The Appleseed Trust Metric*

The Appleseed trust metric constitutes the main contribution of this chapter and is our novel proposal for local group trust metrics. In contrast to Advogato, being inspired by maximum network flow computation, the basic intuition of Appleseed is motivated by *spreading activation models*. Spreading activation models have first been proposed by Quillian [32] in order to simulate human comprehension through semantic memory, and are commonly described as "models of retrieval from long-term memory in which activation subdivides among paths emanating from an activated mental representation" [37]. By the time of this writing, the seminal work of Quillian has been ported to a whole plethora of other disciplines, such as latent semantic indexing

[4] With respect to seed node $a$.

[5] and text illustration [16]. As an example, we will briefly introduce the spreading activation approach adopted in [5], used for semantic search in contextual network graphs, in order to then relate Appleseed to that work.

#### 3.2.2.1 Searches in Contextual Network Graphs

The graph model underlying contextual network search graphs is almost identical in structure to the one presented in Sect. 3.1.1, i.e., edges $(x, y) \in E \subseteq A \times A$ connecting nodes $x, y \in A$. Edges are assigned continuous weights through $W : E \rightarrow [0, 1]$. Source node $s$, the node from which we start searching, is activated through an injection of energy $e$, which is then propagated to other nodes along edges according to some set of simple rules: all energy is *fully divided* among successor nodes with respect to their normalized local edge weight, i.e., the higher the weight of an edge $(x, y) \in E$, the higher the portion of energy that flows along that edge. Furthermore, supposing average outdegrees greater than one, the closer node $x$ to the injection source $s$, and the more paths lead from $s$ to $x$, the higher the amount of energy flowing into $x$. To eliminate endless, marginal and negligible flow, energy streaming into node $x$ must exceed threshold $T$ in order not to run dry. The described approach is captured formally by Algorithm 3.2, which propagates energy recursively.

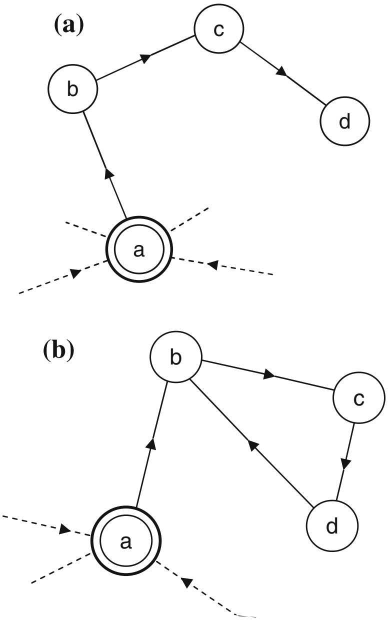
#### 3.2.2.2 Trust Propagation

Algorithm 3.2 shows the basic intuition behind spreading activation models. In order to tailor these models to trust computation, later to become the Appleseed trust metric, serious adaptations are necessary. For instance, procedure energize$(e, s)$ registers *all* energy $e$ that has passed through node $x$, stored in energy$(x)$. Hence, energy$(x)$ represents the *relevance rank* of $x$. Higher values indicate higher node rank. However, at the same time, all energy contributing to the rank of $x$ is passed *without loss* to successor nodes. Interpreting energy ranks as trust ranks thus implies numerous issues of semantic consistency as well as computability. Consider the graph depicted in Fig. 3.4a. Applying spreading activation according to [5], trust ranks of nodes $b$ and $d$ will be identical. However, intuitively, $d$ should be accorded *less* trust than $b$,

```
procedure energize ($e \in \mathbb{R}_0^+, s \in A$) {
  energy($s$) $\leftarrow$ energy($s$) $+ e$;
  $e' \leftarrow e \ / \ \sum_{(s,n) \in E} W(s, n)$;
  if $e > T$ then
    $\forall (s, n) \in E$ : energize ($e' \cdot W(s, n), n$);
  end if
}
```

**Algorithm 3.2**. Recursive energy propagation

**Fig. 3.4** Node chains (**a**) rank sinks (**b**)



since $d$'s shortest-path distance to the trust seed is higher. Trust decay is commonly agreed upon [14, 17], for people tend to trust individuals trusted by immediate friends more than individuals trusted only by friends of friends. Figure 3.4b depicts even more serious issues: all energy, or trust,[5] respectively, distributed along edge $(a, b)$ becomes *trapped in a cycle* and will never be accorded to any other nodes but those being part of that cycle, i.e., $b$, $c$, and $d$. These nodes will eventually acquire infinite trust rank. Obviously, the *bottleneck property* [22] does not hold. Similar issues occur with simplified versions of PageRank [31], where cycles accumulating infinite rank have been dubbed "rank sinks".

### 3.2.2.3 Spreading Factor

We handle both issues, i.e., trust decay in node chains and elimination of rank sinks, by tailoring the algorithm to rely upon our global *spreading factor* $d$. Hereby, let $\text{in}(x)$ denote the energy influx into node $x$. Parameter $d$ then denotes the portion of energy $d \cdot \text{in}(x)$ that node $x$ distributes among successors, while retaining $(1-d) \cdot \text{in}(x)$. For instance, suppose $d = 0.85$ and energy quantity $\text{in}(x) = 5.0$ flowing into node $x$. Then, the total energy distributed to successor nodes amounts to 4.25, while the energy rank $\text{energy}(x)$ of $x$ increases by 0.75. Special treatment is necessary for

[5] The terms "energy" and "trust" are used interchangeably in this context.

nodes with zero outdegree. For simplicity, we assume all nodes to have an outdegree of at least one, which makes perfect sense, as will be shown later.

The spreading factor concept is very intuitive and, in fact, very close to real models of energy spreading through networks. Observe that the overall amount of energy in the network, after initial activation $\text{in}^0$, does not change over time. More formally, suppose that $\text{energy}(x) = 0$ for all $x \in A$ before injection $\text{in}^0$ into source $s$. Then the following equation holds in every computation step of our modified spreading algorithm, incorporating the concept of spreading factor $d$:

$$\sum_{x \in A} \text{energy}(x) = \text{in}^0 \tag{3.2}$$

Spreading factor $d$ may also be seen as the *ratio* between *direct* trust in $x$ and trust in the ability of $x$ to *recommend* others as trustworthy peers. For instance, Beth et al. [4] and Maurer [28] explicitly differentiate between *direct* trust edges and *recommendation* edges.

We commonly assume $d = 0.85$, though other values may also seem reasonable. For instance, having $d \leq 0.5$ allows agents to keep most of the trust they are granted for themselves and only pass small portions of trust to their peers. Observe that low values for $d$ favor trust proximity to the source of trust injection, while high values allow trust to also reach more distant nodes. Furthermore, the introduction of spreading factor $d$ is crucial for making Appleseed retain Levien's bottleneck property, as will be shown in later sections.

#### 3.2.2.4 Rank Normalization

Algorithm 3.2 makes use of edge weight normalization, i.e., the quantity $e_{x \rightarrow y}$ of energy distributed along $(x, y)$ from $x$ to successor node $y$ depends on the *relative* weight of $x \rightarrow y$, i.e., $W(x, y)$ compared to the sum of weights of all outgoing edges of $x$:

$$e_{x \rightarrow y} = d \cdot \text{in}(x) \cdot \frac{W(x, y)}{\sum_{(x,s) \in E} W(x, s)} \tag{3.3}$$

Normalization is common practice in many trust metrics, among those PageRank [31], EigenTrust [18], and AORank [14]. However, while normalized reputation or trust seem reasonable for models with plain, non-weighted edges, serious interferences occur when edges are *weighted*, as is the case for our trust model adopted in Sect. 3.1.1.

For instance, refer to Fig. 3.5a for unwanted effects: The amounts of energy that node $a$ accords to successors $b$ and $d$, i.e., $e_{a \rightarrow b}$ and $e_{a \rightarrow d}$, respectively, are identical in value. Note that $b$ has issued only *one* trust statement $W(b, c) = 0.25$, stating that $b$'s trust in $c$ is rather weak. On the other hand, $d$ assigns *full* trust to individuals $e$, $f$, and $g$. Nevertheless, the overall trust rank for $d$ will be much higher than for any successor of $d$, for $c$ is accorded $e_{a \rightarrow b} \cdot d$, while $e$, $f$, and $g$ only obtain $e_{a \rightarrow d} \cdot d \cdot 1/3$

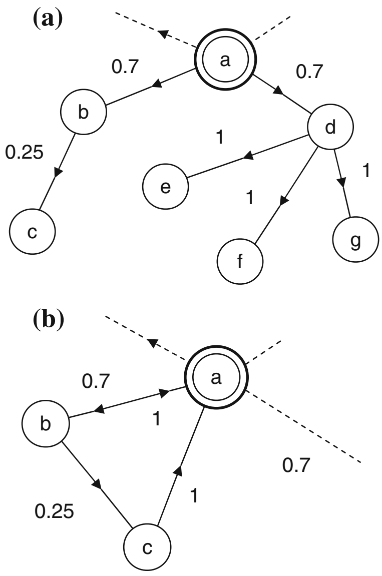


**Fig. 3.5** Issues with trust normalization

each. Hence, $c$ will be trusted *three times* as much as $e$, $f$, and $g$, which is not reasonable at all.

### 3.2.2.5 Backward Trust Propagation

The above issue has already been discussed by Kamvar et al. [18], but no solution has been proposed therein, arguing that "substantially good results" have been achieved despite the drawbacks. We propose to alleviate the problem by making use of *backward propagation* of trust to the source: when metric computation takes place, additional "virtual" edges $(x, s)$ from every node $x \in A \setminus \{s\}$ to the trust source $s$ are created. These edges are assigned full trust $W(x, s) = 1$. Existing backward links $(x, s)$, along with their weights, are "overwritten". Intuitively, every node is supposed to *blindly trust the trust source* $s$, see Fig. 3.5b. The impacts of adding backward propagation links are threefold:

*Mitigating relative trust.* Again, we refer to Fig. 3.5a. Trust distribution in the underlying case becomes much fairer through backward propagation links, for $c$ now only obtains $e_{a \to b} \cdot d \cdot (0.25/(1 + 0.25))$ from source $s$, while $e$, $f$, and $g$ are accorded $e_{a \to d} \cdot d \cdot (1/4)$ each. Hence, trust ranks of both $e$, $f$, and $g$ amount to $1.25$ times the trust assigned to $c$.

*Avoidance of dead ends.* Dead ends, i.e., nodes $x$ with zero outdegree, require special treatment in our computation scheme. Two distinct approaches may be adopted. First, the portion of incoming trust $d \cdot \text{in}(x)$ supposed to be passed to successor nodes is completely discarded, which contradicts our intuition of no energy leaving the system. Second, instead of retaining $(1 - d) \cdot \text{in}(x)$ of incoming trust, $x$ keeps *all* trust. The latter approach is also not sensible as it encourages users to not issue trust statements for their peers. Luckily, with backward propagation of trust, all nodes are *implicitly linked* to the trust source $s$, so that there are no more dead ends to consider.

*Favoring trust proximity.* Backward links to the trust source $s$ are favorable for nodes close to the source, as their eventual trust rank will increase. On the other hand, nodes further away from $s$ are penalized.

#### 3.2.2.6 Nonlinear Trust Normalization

In addition to backward propagation, we propose supplementary measures to decrease the negative impact of trust spreading based on relative weights. Situations where nodes $y$ with poor ratings from $x$ are awarded high overall trust ranks, thanks to the low outdegree of $x$, have to be avoided. Taking the squares of local trust weights provides an appropriate solution:

$$e_{x \rightarrow y} = d \cdot \text{in}(x) \cdot \frac{W(x, y)^2}{\sum_{(x,s) \in E} W(x, s)^2} \tag{3.4}$$

As an example, refer to node $b$ in Fig. 3.5b. With squared normalization, the total amount of energy flowing backward to source $a$ increases, while the amount of energy flowing to the poorly trusted node $c$ decreases significantly. Accorded trust quantities $e_{b \rightarrow a}$ and $e_{b \rightarrow c}$ amount to $d \cdot \text{in}(b) \cdot (1/1.0625)$ and $d \cdot \text{in}(b) \cdot (0.0625/1.0625)$, respectively. A more severe penalization of poor trust ratings can be achieved by selecting powers above two.

#### 3.2.2.7 Algorithm Outline

Having identified modifications to apply to spreading activation models in order to tailor them for local group trust metrics, we are now able to formulate the core algorithm of Appleseed. Input and output are characterized as follows:

$$\text{Trust}_{\alpha} : A \times \mathbb{R}_0^+ \times [0, 1] \times \mathbb{R}^+ \rightarrow (\text{trust} : A \rightarrow \mathbb{R}_0^+) \tag{3.5}$$

The first input parameter specifies trust seed $s$, the second trust injection $e$, parameter three identifies spreading factor $d \in [0, 1]$, and the fourth argument binds accuracy threshold $T_c$, which serves as convergence criterion. Similar to Advogato, the output is an assignment function of trust with domain $A$. However, Appleseed allows *rankings* of agents with respect to trust accorded. Advogato, on the other hand, only assigns boolean values indicating presence or absence of trust.

Appleseed works with *partial* trust graph information. Nodes are accessed only when needed, i.e., when reached by energy flow. Trust ranks trust$(x)$, which correspond to energy$(x)$ in Algorithm 3.2, are initialized to 0. Any unknown node $u$ hence obtains trust$(u) = 0$. Likewise, virtual trust edges for backward propagation from node $x$ to the source are added *at the moment that $x$ is discovered*. In every iteration, for those nodes $x$ reached by flow, the amount of incoming trust is computed as follows:

$$\text{in}(x) = d \cdot \sum_{(p,x) \in E} \left( \text{in}(p) \cdot \frac{W(p,x)}{\sum_{(p,s) \in E} W(p,s)} \right) \tag{3.6}$$

Incoming flow for $x$ is hence determined by all flow that predecessors $p$ distribute along edges $(p, x)$. Note that the above equation makes use of *linear normalization* of relative trust weights. The replacement of linear by nonlinear normalization according to Sect. 3.2.2.6 is straight-forward, though. The trust rank of $x$ is updated as follows:

$$\text{trust}(x) \leftarrow \text{trust}(x) + (1 - d) \cdot \text{in}(x) \tag{3.7}$$

Trust networks generally contain cycles and thus allow no topological sorting of nodes. Hence, the computation of in$(x)$ for reachable $x \in A$ becomes *inherently recursive*. Several iterations for all nodes are required in order to make the computed information converge towards the least fixpoint. The following criterion has to be satisfied for convergence, relying upon accuracy threshold $T_c$ briefly introduced before.

**Definition 3.1** (*Termination*) Suppose that $A_i \subseteq A$ represents the set of nodes that were discovered until step $i$, and trust$_i(x)$ the current trust ranks for all $x \in A$. Then the algorithm terminates when the following condition is satisfied after step $i$:

$$\forall x \in A_i : \text{trust}_i(x) - \text{trust}_{i-1}(x) \leq T_c \tag{3.8}$$

Informally, Appleseed terminates when changes of trust ranks with respect to the preceding iteration $i - 1$ are not greater than accuracy threshold $T_c$.

Moreover, when supposing spreading factor $d > 0$, accuracy threshold $T_c > 0$, and trust source $s$ part of some connected component $G' \subseteq G$ containing at least two nodes, convergence, and thus termination, is guaranteed. The following paragraph gives an informal proof:

*Proof* (*Convergence of Appleseed*) Assume that $f_i$ denotes step $i$'s quantity of energy flowing through the network, i.e., all the trust that has not been captured by some node $x$ through function trust$_i(x)$. From Eq. 3.2 follows that in$^0$ constitutes the *upper boundary* of trust energy floating through the network, and $f_i$ can be computed as follows:

$$f_i = \text{in}^0 - \sum_{x \in A} \text{trust}_i(x) \tag{3.9}$$

Since $d > 0$ and $\exists (s, x) \in E, x \neq s$, the sum of the current trust ranks trust$_i(x)$ of all $x \in A$ is *strictly increasing* for increasing $i$. Consequently, $\lim_{i \to \infty} f_i = 0$ holds.

```
**func** Trust$_\alpha$ ($s \in A$, in$^0 \in \mathbb{R}_0^+$, $d \in [0, 1]$, $T_c \in \mathbb{R}^+$) {
  set in$_0(s) \leftarrow$ in$^0$, trust$_0(s) \leftarrow 0$, $i \leftarrow 0$;
  set $A_0 \leftarrow \{s\}$;
  **repeat**
    set $i \leftarrow i + 1$;
    set $A_i \leftarrow A_{i-1}$;
    $\forall x \in A_{i-1}$ : set in$_i(x) \leftarrow 0$;
    **for all** $x \in A_{i-1}$ **do**
      set trust$_i(x) \leftarrow$ trust$_{i-1}(x) + (1 - d) \cdot$ in$_{i-1}(x)$;
      **for all** $(x, u) \in E$ **do**
        **if** $u \notin A_i$ **then**
          set $A_i \leftarrow A_i \cup \{u\}$;
          set trust$_i(u) \leftarrow 0$, in$_i(u) \leftarrow 0$;
          add edge $(u, s)$, set $W(u, s) \leftarrow 1$;
        **end if**
        set $w \leftarrow W(x, u) \,/\, \sum_{(x,u') \in E} W(x, u')$;
        set in$_i(u) \leftarrow$ in$_i(u) + d \cdot$ in$_{i-1}(x) \cdot w$;
      **end do**
    **end do**
    set $m = \max_{y \in A_i}\{$trust$_i(y) -$ trust$_{i-1}(y)\}$;
  **until** $(m \leq T_c)$
  **return** (trust : $\{(x,$ trust$_i(x)) \mid x \in A_i\}$);
}
```

**Algorithm 3.3**. Outline of the Appleseed trust metric

Moreover, since termination is defined by some fixed accuracy threshold $T_c > 0$, there exists some step $k$ such that $\lim_{i \to k} f_i \leq T_c$.

#### 3.2.2.8 Parameterization and Experiments

Appleseed allows numerous parameterizations of input variables, some of which are subject to discussion in the section at hand. Moreover, we provide experimental results exposing the observed effects of parameter tuning. Note that all experiments have been conducted on data obtained from "real" social networks: we have written several Web crawling tools to mine the Advogato community Web site and extract trust assertions stated by its more than 8,000 members.[6] Hereafter, we converted all trust data to our trust model proposed in Sect. 3.1.1. The Advogato community server supports four different levels of peer certification, namely Observer, Apprentice, Journeyer, and Master. We mapped these *qualitative* certification levels to quantitative ones, assigning $W(x, y) = 0.25$ for $x$ certifying $y$

[6] Crawls have been executed in September 2004.

as OBSERVER, $W(x, y) = 0.5$ for an APPRENTICE, and so forth. The Advogato community undergoes rapid growth and our crawler extracted 3,224,101 trust assertions. Preprocessing and data cleansing were thus inevitable, eliminating reflexive trust statements $W(x, x)$ and shrinking trust certificates to reasonable sizes. Note that some eager Advogato members have issued *more than two thousand* trust statements, yielding an overall average outdegree of 397.69 assertions per node.Clearly, this figure is beyond dispute. Hence, applying our set of extraction tools, we tailored the test data obtained from Advogato to our needs and extracted trust networks with specific average outdegrees for the experimental analysis.

### Trust Injection

Trust values $\text{trust}(x)$ computed by the Appleseed metric for source $s$ and node $x$ may differ greatly from explicitly assigned trust weights $W(s, x)$. We already mentioned before that computed trust ranks may *not* be interpreted as absolute values, but rather in comparison with ranks assigned to all other peers. In order to make assigned rank values more tangible, though, one might expect that tuning the trust injection $\text{in}^0$ to satisfy the following proposition will align computed ranks and explicit trust statements:

$$\forall (s, x) \in E : \ \text{trust}(x) \in [W(s, x) - \varepsilon, W(s, x) + \varepsilon] \tag{3.10}$$

However, when assuming reasonably small $\varepsilon$, the approach does not succeed. Recall that *computed* trust values of successor nodes $x$ of $s$ do not only depend on assertions made by $s$, but also on trust ratings asserted by other peers. Hence, a perfect alignment of explicit trust ratings with computed ones cannot be accomplished. However, we propose a heuristic alignment method, incorporated into Algorithm 3.4, which has proven to work remarkably well in diverse test scenarios. The basic idea is to add another node $i$ and edge $(s, i)$ with $W(s, i) = 1$ to the trust graph $G = (A, E, W)$, treating $(s, i)$ as an indicator to test whether trust injection $\text{in}^0$ is "good" or not. Consequently, parameter $\text{in}^0$ has to be adapted in order to make $\text{trust}(i)$ converge

```
func Trust_heu ($s \in A, d \in [0, 1], T_c \in \mathbb{R}^+$) {
  add node $i$, edge $(s, i)$, set $W(s, i) \leftarrow 1$;
  set $\text{in}^0 \leftarrow 20, \varepsilon \leftarrow 0.1$;
  repeat
    set trust $\leftarrow$ Trust_α ($s, \text{in}^0, d, T_c$);
    $\text{in}^0 \leftarrow$ adapt ($W(s, i)$, trust($i$), $\text{in}^0$);
  until trust($i$) $\in [W(s, i) - \varepsilon, W(s, i) + \varepsilon]$
  remove node $i$, remove edge (s, i);
  return Trust_α ($s, \text{in}^0, d, T_c$);
}
```

**Algorithm 3.4**. Heuristic weight alignment method towards $W(s, i)$. The trust metric computation is hence repeated with different values for $\text{in}^0$ until convergence of the explicit and the computed trust value of $i$ is achieved. Eventually, edge $(s, i)$ and node $i$ are removed and the computation is performed one more time. Experiments have shown that our imperfect alignment method yields computed ranks $\text{trust}(x)$ for direct successors $x$ of trust source $s$ which come close to previously specified trust statements $W(s, x)$.

### Spreading Factor

Small values for $d$ tend to overly reward nodes close to the trust source and penalize remote ones. Recall that *low d* allows nodes to retain most of the incoming trust quantity for themselves, while *large d* stresses the recommendation of trusted individuals and makes nodes distribute most of the assigned trust to their successor nodes.

**Experiment 1 (Spreading factor impact)** We compare distributions of computed rank values for three diverse instantiations of $d$, namely $d_1 = 0.1$, $d_2 = 0.5$, and $d_3 = 0.85$. Our setup is based upon a social network with an average outdegree of 6 trust assignments, and features 384 nodes reached by trust energy spreading from our designated trust source. We furthermore suppose $\text{in}^0 = 200$, $T_c = 0.01$, and *linear* weight normalization. Computed ranks are classified into 11 histogram cells with nonlinear cell width. Obtained output results are displayed in Fig. 3.6. Mind that we have chosen *logarithmic* scales for the vertical axis in order to render the diagram more legible. For $d_1$, we observe that the largest number of nodes $x$ with ranks $\text{trust}(x) \geq 25$ is generated. On the other hand, virtually no ranks ranging from 0.2 to 1 are assigned, while the number of nodes with ranks smaller than 0.05 is again much higher for $d_1$ than for both $d_2$ and $d_3$. Instantiation $d_3 = 0.85$ exhibits behavior opposed to that of $d_1$. No ranks with $\text{trust}(x) \geq 25$ are accorded, while interim ranks between 0.1 and 10 are much more likely for $d_3$ than for both other instantiations of spreading factor $d$. Consequently, the number of ranks below 0.05 is lowest for $d_3$.

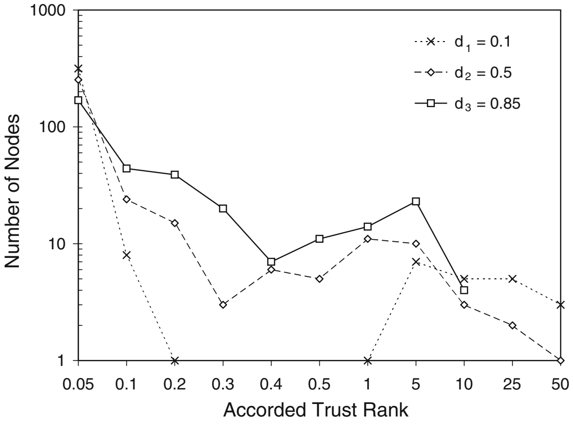


**Fig. 3.6** Spreading factor impact

The experiment demonstrates that high values for parameter $d$ tend to distribute trust more evenly, neither overly rewarding nodes close to the source, nor penalizing remote ones too rigidly. On the other hand, low $d$ assigns high trust ranks to very few nodes, namely those which are closest to the source, while the majority of nodes obtains very low trust rank. We propose to set $d = 0.85$ for general use.

### Convergence

We already mentioned before that the Appleseed algorithm is *inherently recursive*. Parameter $T_c$ represents the ultimate criterion for termination. We demonstrate through an experiment that convergence is reached very fast, no matter how large the number of nodes trust is flowing through, and no matter how large the initial trust injection.

**Experiment 2 (Convergence rate)** The trust network we consider has an average outdegree of 5 trust statements per node. The number of nodes for which trust ranks are assigned amounts to 572. We suppose $d = 0.85$, $T_c = 0.01$, and *linear* weight normalization. Two separate runs were computed, one with trust activation $\text{in}_1 = 200$, the other with initial energy $\text{in}_2 = 800$. Figure 3.7 demonstrates the rapid convergence of both runs. Though the trust injection for the second run is 4 times as high as for the first, convergence is reached in only few more iterations: run one takes 38 iterations, run two terminates after 45 steps.

For both runs, we assumed accuracy threshold $T_c = 0.01$, which is extremely small and accurate beyond necessity already. However, experience taught us that convergence takes place rapidly even for very large networks and high amounts of trust injected, so that assuming the latter value for $T_c$ poses no scalability issues. In fact, the amount of nodes taken into account for trust rank assignment in the

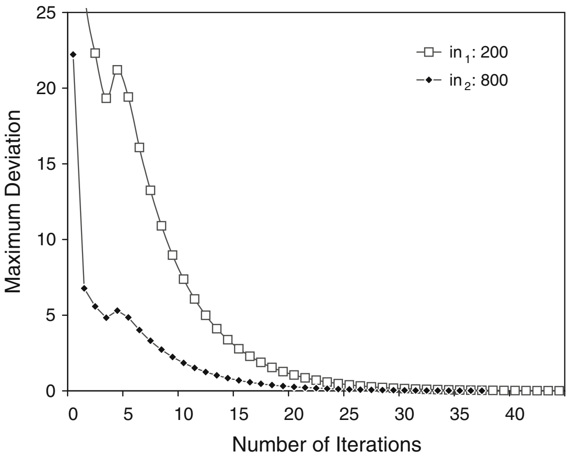


**Fig. 3.7** Convergence of Appleseed

above example well exceeds practical usage scenarios: mind that the case at hand demands 572 files to be fetched from the Web, complaisantly supposing that these pages are cached after their first access. Hence, we claim that the actual bottleneck of group trust computation is *not* the Appleseed metric itself, but downloads of trust resources from the network. This bottleneck might also be the reason for selecting thresholds $T_c$ greater than 0.01, in order to make the algorithm terminate after fewer node accesses.

#### Testbed Design and Experimental Trials

Trust metrics and models for trust propagation have to be *intuitive*, i.e., humans must eventually comprehend *why* agent $a_i$ has been accorded a higher trust rank than $a_j$ and come to similar results when asked for personal judgement. Consequently, we implemented our own testbed, which graphically displays social networks. We made use of the YFILES [39] library to perform complex graph drawing and layouting tasks. The testbed allows for parameterizing Appleseed through dialogs. Detailed output is provided, both graphical and textual. Graphical results comprise the highlighting of nodes with trust ranks above certain thresholds, while textual results return quantitative trust ranks of all accessed nodes, the number of iterations, and so forth. We also implemented the Advogato trust metric and incorporated the latter into our testbed. Hereby, our implementation of Advogato does not require a priori complete trust graph information, but accesses nodes "just in time", similar to Appleseed. All experiments were conducted on top of the testbed application.

### *3.2.3 Comparison of Advogato and Appleseed*

Advogato and Appleseed are both implementations of local group trust metrics. Advogato has already been successfully deployed into the Advogato online community, though quantitative evaluation results have not been provided yet. In order to evaluate the fitness of Appleseed as an appropriate means for group trust computation, we relate our approach to Advogato for qualitative comparison:

(F.1) *Attack-resistance.* This property defines the behavior of trust metrics in case of malicious nodes trying to invade into the system. For evaluation of attack-resistance capabilities, we have briefly introduced the "bottleneck property" in Sect. 3.2.1.2, which holds for Advogato. In order to recapitulate, suppose that $s$ and $t$ are nodes and connected through trust edge $(s, t)$. Node $s$ is assumed good, while $t$ is an attacking agent trying to make good nodes trust malevolent ones. In case the bottleneck property holds, manipulation "on the part of bad nodes does not affect the trust value" [22]. Clearly, Appleseed satisfies the bottleneck property, for nodes cannot raise their impact by modifying the structure of trust statements they issue. Bear in mind that the amount of trust accorded to agent $t$ *only* depends on his predecessors and does not increase when $t$ adds more nodes. Both, spreading factor $d$ and normalization of trust statements, ensure that Appleseed maintains attack-resistance properties according to Levien's definition.

(F.2) *Eager truster penalization.* We have indicated before that issuing multiple trust statements dilutes trust accorded to successors. According to Guha [14], this does not comply with real world observations, where statements of trust "do not decrease in value when the user trusts one more person [...]". The malady that Appleseed suffers from is common to many trust metrics, most notably those based upon finding principal eigenvectors [18, 31, 35]. On the other hand, the approach pursued by Advogato does *not* penalize trust relationships asserted by eager trust dispensers, for node capacities do not depend on *local* information. Remember that capacities of nodes pertaining to level $l$ are assigned based on the capacity of level $l - 1$, as well as the *overall* outdegree of nodes part of that level. Hence, Advogato *encourages* agents issuing numerous trust statements, while Appleseed *penalizes* overly abundant trust certificates.

(F.3) *Deterministic trust computation.* Appleseed is deterministic with respect to the assignment of trust rank to agents. Hence, for any arbitrary trust graph $G = (A, E, W)$ and for every node $x \in A$, linear equations allow for characterizing the amount of trust assigned to $x$, as well as the quantity that $x$ accords to successor nodes. Advogato, however, is *non-deterministic*. Though the *number* of trusted agents, and therefore the computed maximum flow size, is determined for given input parameters, the set of agents is not. Changing the order in which trust assertions are issued may yield different results. For example, suppose $C_A(s) = 1$ holds for trust seed $s$. Furthermore, assume $s$ has issued trust certificates for two agents, $b$ and $c$. The actual choice between $b$ or $c$ as trustworthy peer with maximum flow *only depends on the order* in which nodes are accessed.

(F.4) *Model and output type.* Basically, Advogato supports non-weighted trust statements only. Appleseed is more versatile by virtue of its trust model based on *weighted* trust certificates. In addition, Advogato returns one set of trusted peers, whereas Appleseed assigns *ranks* to agents. These ranks allow to select most trustworthy agents first and relate them to each other with respect to their accorded rank. Hereby, the definition of thresholds for trustworthiness is left to the user who can thus tailor relevant parameters to fit different application scenarios. For instance, raising the application-dependent threshold for the selection of trustworthy peers, which may be either an absolute or a relative value, allows for enlarging the neighborhood of trusted peers. Appleseed is hence more adaptive and flexible than Advogato.

The afore-mentioned characteristics of Advogato and Appleseed are briefly summarized in Table 3.1.

**Table 3.1** Characteristics of Advogato and Appleseed

| | Feature **F.1** | Feature **F.2** | Feature **F.3** | Feature **F.4** |
|---|---|---|---|---|
| Advogato | Yes | No | No | Boolean |
| Appleseed | Yes | Yes | Yes | Ranking |

## 3.3 Distrust

The notion of distrust is one of the most controversial topics when defining trust metrics and trust propagation. Most approaches completely *ignore* distrust and only consider *full* trust or *degrees of trust* [4, 23, 28, 30, 33, 35]. Others, among those [1, 3, 6, 13], allow for distrust ratings, though, but do not consider the subtle semantic differences that exist between those two notions, i.e., trust and distrust. Consequently, according to [9], "distrust is regarded as just the other side of the coin, that is, there is generally a symmetric scale with complete trust on one end and absolute distrust on the other". Furthermore, some researchers equate the notion of distrust with *lack of trust information*. However, in his seminal work on the essence of trust, Marsh [26] has already pointed out that those two concepts, i.e., lack of trust and distrust, may *not* be intermingled. For instance, in absence of trustworthy agents, one might be more prone to accept recommendations from non-trusted persons, being non-trusted probably because of lack of prior experiences [27], than from persons we explicitly *distrust*, the distrust resulting from bad past experiences or deceit. However, even Marsh pays little attention to the specifics of distrust.

Gans et al. [9] were among the first to recognize the importance of distrust, stressing the fact that "distrust is an irreducible phenomenon that cannot be offset against any other social mechanisms", including trust. In their work, an explicit distinction between confidence, trust, and distrust is made. Moreover, the authors indicate that distrust might be highly relevant to social networks. Its impact is not inherently negative, but may also influence the network in an extremely positive fashion. However, the primary focus of this work is on methodology issues and planning, not considering trust assertion evaluations and propagation through appropriate metrics.

Guha et al. [15] acknowledge the immense role of distrust with respect to trust propagation applications, arguing that "distrust statements are very useful for users to debug their web of trust" [14]. For example, suppose that agent $a_i$ blindly trusts $a_j$, which again blindly trusts $a_k$. However, $a_i$ completely distrusts $a_k$. The distrust statement hence ensures that $a_i$ will *not* accept beliefs and ratings from $a_k$, irrespective of him trusting $a_j$ trusting $a_k$.

### *3.3.1 Semantics of Distrust*

The non-symmetrical nature of distrust and trust, being two dichotomies, has already been recognized by recent sociological research [25]. In this section, we investigate the differences between distrust and trust with respect to inference opportunities and the propagation of beliefs.

#### 3.3.1.1 Distrust as Negated Trust

Interpreting distrust as the negation of trust has been adopted by many trust metrics, among those trust metrics proposed by Abdul-Rahman and Hailes [1, 2], Jøsang et al. [17], and Chen and Yeager [6]. Basically, these metrics compute trust values by analyzing *chains* of trust statements from source $s$ to target $t$, eventually merging them to obtain an aggregate value. Each chain hereby becomes synthesized into one single number through *weighted multiplication* of trust values along trust paths. Serious implications resulting from the assumption that trust concatenation relates to multiplication [35], and distrust to negated trust, arise when agent $a_i$ distrusts $a_j$, who distrusts $a_k$[7]:

$$\neg\,\mathrm{trust}(a_i, a_j) \wedge \neg\,\mathrm{trust}(a_j, a_k) \models \mathrm{trust}(a_i, a_k) \tag{3.11}$$

Jøsang et al. [17] are aware of this rather unwanted effect, but do not question its correctness, arguing that "the enemy of your enemy could well be your friend". Guha [14], on the other hand, indicates that two distrust statements canceling out each other commonly does *not* reflect desired behavior.

#### 3.3.1.2 Propagation of Distrust

The *conditional transitivity* of trust [1] is commonly agreed upon and represents the foundation and principal premiss that trust metrics rely upon. However, no consensus in literature has been achieved with respect to the *degree* of transitivity and the decay rate of trust. Many approaches therefore explicitly distinguish between *recommendation* trust and *direct* trust [1, 4, 6, 17, 28] in order to keep apart the transitive fraction of trust from the non-transitive. Hence, in these works, only the *ultimate* edge within the trust chain, i.e., the one linking to the trust target, needs to be direct, while all others are supposed to be recommendations. For the Appleseed trust metric, this distinction is made through the introduction of spreading factor $d$. However, the conditional transitivity property of trust does not equally extend to distrust. The case of double negation through distrust propagation has already been considered. Now suppose, for instance, that $a_i$ distrusts $a_j$, who trusts $a_k$. Supposing distrust to propagate through the network, we come to make the following inference:

$$\mathrm{distrust}(a_i, a_j) \wedge \mathrm{trust}(a_j, a_k) \models \mathrm{distrust}(a_i, a_k) \tag{3.12}$$

The above inference is more than questionable, for $a_i$ penalizes $a_k$ simply for being trusted by an agent $a_j$ that $a_i$ distrusts. Obviously, this assumption is not sound and does not reflect expected real-world behavior. We assume that distrust does not allow for making direct inferences *of any kind*. This conservative assumption well complies with [14].

[7] We oversimplify by using predicate calculus expressions, supposing that trust, and hence distrust, is fully transitive.

## 3.3.2 *Incorporating Distrust into Appleseed*

We compare our distrust model with Guha's approach, making similar assumptions. Guha computes trust by means of *one global* group trust metric, similar to PageRank [31]. For distrust, he proposes two candidate approaches. The first one directly integrates distrust into the iterative eigenvector computation and comes up with one single measure combining both trust and distrust. However, in networks dominated by distrust, the iteration might not converge [14]. The second proposal first computes trust ranks by trying to find the dominant eigenvector, and then computes separate distrust ranks in one single step, based upon the iterative computation of trust ranks. Suppose that $D_{a_i}$ is the set of agents who distrust $a_i$:

$$\text{DistrustRank}(a_i) = \frac{\sum_{a_j \in D_{a_i}} \text{TrustRank}(a_j)}{|D_{a_i}|} \tag{3.13}$$

The problem we perceive with this approach refers to *superimposing* the computation of distrust ranks *after* trust rank computation, which may yield some strange behavior: suppose an agent $a_i$ who is highly controversial by engendering ambiguous sentiments, i.e., on the one hand, there are numerous agents that *trust* $a_i$, while on the other hand, there are numerous agents who *distrust* $a_i$. With the approach proposed by Guha, $a_i$'s impact for distrusting other agents is huge, resulting from his immense positive trust rank. However, this should clearly not be the case, for $a_i$ is subject to tremendous distrust himself, thus leveling out his high trust rank.

Hence, for our own approach, we intend to *directly* incorporate distrust into the iterative process of the Appleseed trust metric computation, and not superimpose distrust afterwards. Several pitfalls have to be avoided, such as the risk of non-convergence in case of networks dominated by distrust [14]. Furthermore, in absence of distrust statements, we want the distrust-enhanced Appleseed algorithm, which we denote by $\text{Trust}_{\alpha^-}$, to yield results identical to those engendered by the original version $\text{Trust}_{\alpha}$.

#### 3.3.2.1 Normalization and Distrust

First, the trust normalization procedure has to be adapted. We suppose normalization of weights to the power of $q$, as has been discussed in Sect. 3.2.2.6. Let $\text{in}(x)$, the trust influx for agent $x$, be *positive*. As usual, we denote the global spreading factor by $d$, and quantified trust statements from $x$ to $y$ by $W(x, y)$. Function $\text{sign}(x)$ returns the sign of value $x$. Note that from now on, we assume $W : E \to [-1, +1]$, for degrees of *distrust* need to be expressible. Then the trust quantity $e_{x \to y}$ passed from $x$ to successor $y$ is computed as follows:

$$e_{x \to y} = d \cdot \text{in}(x) \cdot \text{sign}(W(x, y)) \cdot w, \tag{3.14}$$

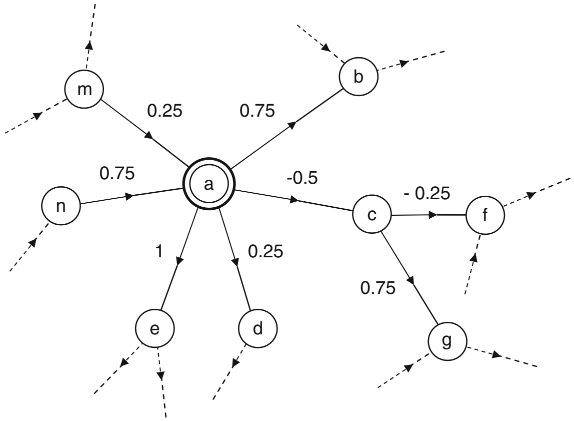


**Fig. 3.8** Network augmented by distrust

where

$$w = \frac{|W(x, y)|^q}{\sum_{(x,s) \in E} |W(x, s)|^q}$$

The accorded quantity $e_{x \to y}$ becomes *negative* if $W(x, y)$ is negative, i.e., if $x$ distrusts $y$. For the relative weighting, the *absolute* values $|W(x, s)|$ of all weights are considered. Otherwise, the denominator could become negative, or positive trust statements could become boosted unduly. The latter would be the case if the sum of positive trust ratings *only slightly* outweighed the sum of negative ones, making the denominator converge towards zero. An example demonstrates the computation process:

*Example 3.2* (Distribution of Trust and Distrust) We assume the trust network as depicted in Fig. 3.8. Let the trust energy influx into node $a$ be $\text{in}(a) = 2$, and global spreading factor $d = 0.85$. For simplicity reasons, backward propagation of trust to the source is *not* considered. Moreover, we suppose *linear* weight normalization, thus $q = 1$. Consequently, the denominator of the normalization equation is $|0.75| + |-0.5| + |0.25| + |1| = 2.5$. The trust energy that $a$ distributes to $b$ hence amounts to $e_{a \to b} = 0.51$, whereas the energy accorded to the distrusted node $c$ is $e_{a \to c} = -0.34$. Furthermore, we have $e_{a \to d} = 0.17$ and $e_{a \to e} = 0.68$.

Observe that trust energy becomes *lost* during distribution, for the sum of energy accorded along outgoing edges of $a$ amounts to 1.02, while 1.7 was provided for distribution. The effect results from the negative trust weight $W(a, c) = -0.5$.

#### 3.3.2.2 Distrust Allocation and Propagation

We now analyze the case where the influx $\text{in}(x)$ for agent $x$ is *negative*. In this case, the trust allocated for $x$ will also be negative, i.e., $\text{in}(x) \cdot (1 - d) < 0$. Moreover, the energy $\mathrm{in}(x) \cdot d$ that $x$ may distribute among successor nodes will be negative as well. The implications are those which have been mentioned in Sect. 3.3.1, i.e., distrust as negation of trust and propagation of distrust. For the first case, refer to node $f$ in Fig. 3.8 and assume $\mathrm{in}(c) = -0.34$, which is derived from Example 3.2. The trusted agent $a$ distrusts $c$ who distrusts $f$. Eventually, $f$ would be accorded $d \cdot (-0.34) \cdot (-0.25)$, which is *positive*. For the second case, node $g$ would be assigned the *negative* trust quantity $d \cdot (-0.34) \cdot (0.75)$, simply for being trusted by $f$, who is distrusted. Both unwanted effects can be avoided by not allowing distrusted nodes to distribute *any energy at all*. Hence, more formally, we introduce a novel function $\mathrm{out}(x)$:

$$\mathrm{out}(x) = \begin{cases} d \cdot \mathrm{in}(x), & \text{if } \mathrm{in}(x) \geq 0 \\ 0, & \text{else} \end{cases} \tag{3.15}$$

This function then has to replace $d \cdot \mathrm{in}(x)$ when computing the energy distributed along edges from $x$ to successor nodes $y$:

$$e_{x \to y} = \mathrm{out}(x) \cdot \mathrm{sign}(W(x, y)) \cdot w, \tag{3.16}$$

where

$$w = \frac{|W(x, y)|^q}{\sum_{(x,s) \in E} |W(x, s)|^q}$$

This design decision perfectly aligns with assumptions made in Sect. 3.3.1 and prevents the inference of unwanted side-effects mentioned before. Furthermore, one can see easily that the modifications introduced *do not affect* the behavior of Algorithm 3.3 when not considering relationships of distrust.

#### 3.3.2.3 Convergence

In networks largely or entirely dominated by distrust, the extended version of Appleseed is still guaranteed to converge. We therefore briefly outline an informal proof, based on Proof 3.2.2.7:

*Proof* (*Convergence in presence of distrust*) Recall that only *positive* trust influx $\mathrm{in}(x)$ becomes propagated, which has been indicated in Section 3.3.2.2. Hence, all we need to show is that the overall quantity of *positive* trust distributed in computation step $i$ cannot be augmented through the presence of distrust statements. In other words, suppose that $G = (A, E, W)$ defines an arbitrary trust graph, containing quantified trust statements, but *no distrust*, i.e., $W : E \to [0, 1]$. Now consider another trust graph $G' = (A, E \cup D, W')$, which contains additional edges $D$, and weight function $W' = W \cup (D \to [-1, 0[)$. Hence, $G'$ augments $G$ by additional distrust edges between nodes taken from $A$. We now perform two parallel computations with the extended version of Appleseed, one operating on $G$ and the other on $G'$. In every step, and for every trust edge $(x, y) \in E$ for $G$, the distributed energy $e_{x \to y}$ is greater or equal to the respective counterpart on $G'$, because the denominator of the fraction given in Eq. 3.16 can only become *greater* through additional distrust outedges. Second, for the computation performed on $G'$, negative energy distributed along edge $(x, y)$ can only *reduce* the trust influx for $y$ and may hence even accelerate convergence.

However, as can be observed from the proof, there exists one serious implication arising from having distrust statements in the network: the overall accorded trust quantity does *not* equal the initially injected energy anymore. Moreover, in networks dominated by distrust, the overall trust energy sum may even be *negative*.

**Experiment 3 (Network impact of distrust)** We observe the number of iterations until convergence is reached, and the overall accorded trust rank of 5 networks. The structures of all these graphs are identical, being composed of 623 nodes with an average indegree and outdegree of 9. The only difference applies to the assigned weights, where the first graph contains no distrust statements at all, while 25 % of all weights are negative for the second, 50 % for the third, and 75 % for the fourth. The fifth graph contains nothing but distrust statements. The Appleseed parameters are identical for all 5 runs, having backward propagation enabled, an initial trust injection $\text{in}^0 = 200$, spreading factor $d = 0.85$, convergence threshold $T_c = 0.01$, *linear* weight normalization, and no upper bound on the number of nodes to unfold. Figure 3.9a clearly demonstrates that the number of iterations until convergence, given on the vertical axis, *decreases* with the proportion of distrust increasing, observable along the horizontal axis. Likewise, the overall accorded trust rank, indicated on the vertical axis of Fig. 3.9b, decreases rapidly with increasing distrust, eventually dropping below zero. The same experiment was repeated for another network with 329 nodes, an average indegree and outdegree of 6, yielding similar results.

The effects observable in Experiment 3 only marginally affect the ranking itself, for trust ranks are interpreted *relative* to each other. Moreover, compensation for lost trust energy may be achieved by boosting the initial trust injection $\text{in}^0$.

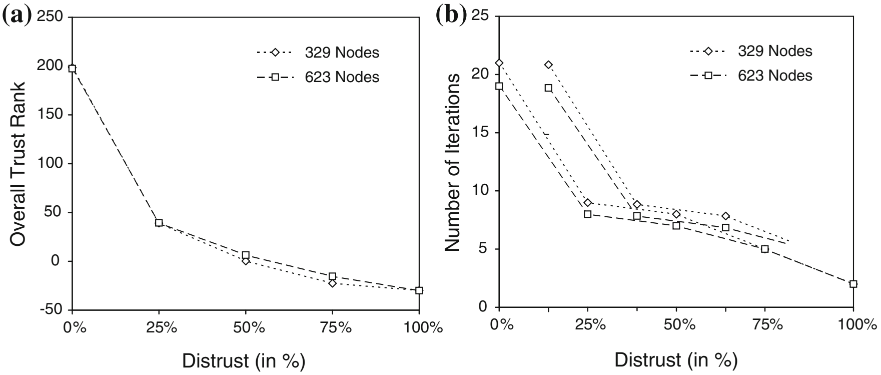


**Fig. 3.9** Network impact of distrust

## 3.4 Expanding Network Coverage

Among the network-based algorithms for computing trust, there is one common problem: coverage. In social networks there are often many users who are disconnected from the main cluster or who are connected in a way that computing an accurate trust value would be difficult. This naturally leads to the question of how we can improve network coverage, and potentially accuracy, in trust computation.

We propose one solution to this which incorporates similarity measures grounded in our sociological understanding of trust. Sociological definitions of trust have two components: a belief and a commitment. For example, in a context, if Alice trusts Bob, it implies that Alice believes that Bob will provide useful information *and* that she is willing to take action based on that information [10]. If we consider this in the context of information on the Web, trust in a person means that the user is willing to take actions, like buying a product, based on others' reviews. This, of course, gets to the core of why we want to compute trust in the first place; if we know how much the user trusts the author of some online content, we can use that to help optimally sort, filter, and aggregate the information.

Network flow-based algorithms, like Appleseed and Advogato presented above work very well on connected graph components, but they cannot infer trust between people who are not connected by paths in the social network. In those situations, trust can be inferred from other sources of information. In our previous work, we identified a series of similarity measures drawn from underlying data that can estimate trust effectively. We will discuss this work further in Sect. 3.4.1. This method can infer trust between any two users as long as they have rated a common set of items upon which to compute similarity. However, computing trust based only on nuanced similarity measures loses some of the insights that come from the network.

In some cases, we will be able to compute trust values with both algorithms for a pair of users. In other cases, we may be able to compute only one (or perhaps neither). A combination of the methods will obviously achieve better coverage, but how to effectively use both values when available is an open question. In this section, we present a model that integrates trust computed from social networks and trust inferred from data similarity. We show results on how to optimally use both models and demonstrate their accuracy on a real-world dataset.

### *3.4.1 Revisiting Trust Inference Algorithms*

Above, we discussed two major trust inference algorithms: Advogato and Appleseed. For the experiments here, we used the TidalTrust [12] algorithm, a simple trust inference algorithm that gives a good indication of how many pairs of users a network-based algorithm can reach. Readers will note similarities between TidalTrust and the algorithms presented above.

#### 3.4.1.1 TidalTrust: An Algorithm for Inferring Trust

TidalTrust is a modified breadth-first search-base algorithm. The source's inferred trust rating for the sink, denoted $t_{s,p}$, is a weighted average of source $s$'s neighbors' ratings of sink $p$. The source node begins a search for the sink. It will poll each of its neighbors to obtain their rating of the sink. If the neighbor has a direct rating of the sink, that value is returned. If the neighbor does not have a direct rating for the sink, it queries all of its neighbors for their ratings, computes the weighted average, and returns the result. Each neighbor repeats this process. Essentially, the nodes perform a breadth-first search from the source to the sink, and then inferred values are passed back to the source. The basic process of values for the sink flowing back to the source are shown in Fig. 3.10.

Network-based inference algorithms rely on the social network. This provides a benefit because recommendations can be made for users who have rated no items because trust is inferred from the social connections. However, it has a corresponding drawback that trust can only be computed when users are connected in that network.

TidalTrust incorporates two factors to limit the size of the search and improve accuracy. Previous research has shown the following [11]:

- Shorter paths have a lower error.
- Using nodes with higher trust ratings leads to lower error.

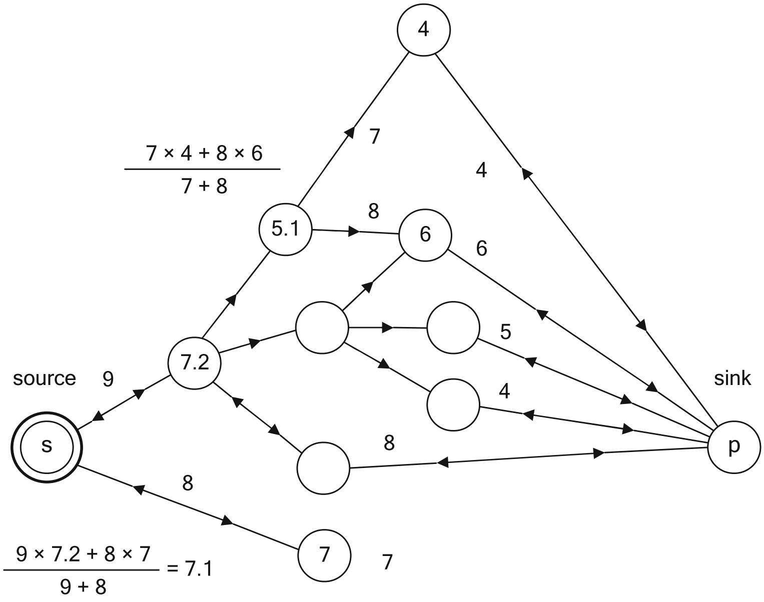

4
7
$\frac{7 \times 4 + 8 \times 6}{7 + 8}$
4
8
5.1
6
6
5
source
9
7.2
sink
4
s
8
p
8
$\frac{9 \times 7.2 + 8 \times 7}{9 + 8} = 7.1$
7
7

**Fig. 3.10** An illustration of direct trust values between nodes $a$ and $b$, $t_{a,b}$, and between nodes $b$ and $c$, $t_{b,c}$. Using a trust inference algorithm, it is possible to compute a value to recommend how much $a$ may trust $c$, $t_{a,c}$

Limiting the depth of TidalTrust's search should lead to more accurate results, since the error often increases as depth increases. If accuracy decreases as path length increases, as the earlier analysis suggests, then shorter paths are more desirable. However, the tradeoff is that fewer nodes will be reachable if a limit is imposed on the path depth. To balance these factors, the path length can vary from one computation to another. Instead of a fixed depth, the shortest path length required to connect the source to the sink becomes the depth. This preserves the benefits of a shorter path length without limiting the number of inferences that can be made.

The previous results also indicate that the most accurate information will come from the highest trusted neighbors. To incorporate this into the algorithm, we establish a minimum trust threshold, and only consider connections in the network with trust ratings at or above the threshold. This value cannot be fixed before the search because we cannot predict what the highest trust value will be along the possible paths. If the value is set too high, some nodes may not have assigned values and no path will be found. If the threshold is too low, then paths with lower trust may be considered when it is not necessary. We define a variable, max, that represents the largest trust value that can be used as a minimum threshold such that a path can be found from source to sink. Our max is computed while searching for paths to the sink by tracking trust values that have been seen.

TidalTrust is a modified breadth-first search. The inferred trust rating of source $s$ for sink $p$, denoted $t_{s,p}$, is a weighted average of the source's neighbors' ratings of the sink. This is succinctly represented as follows:

$$t_{s,p} = \frac{\sum\limits_{j \,\in\, \mathrm{adj}(s) \,\wedge\, t_{s,j} \,\geq\, \max} t_{s,j} \times t_{j,p}}{\sum\limits_{j \,\in\, \mathrm{adj}(s) \,\wedge\, t_{s,j} \,\geq\, \max} t_{s,j}} \tag{3.17}$$

The source node begins a search for the sink. It will poll each of its neighbors to obtain their rating of the sink. If the neighbor has a direct rating of the sink, that value is returned. If the neighbor does not have a direct rating for the sink, it queries all of its neighbors for their ratings, computes the weighted average as shown above, and returns the result .

To improve the accuracy of the algorithm, path length and path strength considerations are included. Each node that is reached performs this process, keeping track of the current depth from the source. Each node will also keep track of the strength of the path to it. Nodes adjacent to the source will record the source's rating assigned to them. Each of those nodes will poll their neighbors. The strength of the path to each neighbor is the minimum of the source's rating of the node and the node's rating of its neighbor. The neighbor records the maximum strength path leading to it. Once a path is found from the source to the sink, the depth is set at the maximum depth allowable. Since the search is proceeding in a breadth-first search fashion, the first path found will be at the minimum depth. The search will continue to find any other paths at the minimum depth. Once this search is complete, the trust threshold (max) is established by taking the maximum of the trust paths leading to the sink. With the max value established, each node can complete the calculations of a weighted average by taking information from nodes that they have rated at or above the max threshold.

The accuracy of this algorithm is addressed in depth in [10, 12]. While the error will very from network to network, our experiments in two real world social networks show the results to be accurate to within about 10 %.

#### 3.4.1.2 Similarity-Based Trust Inference

It has been long known in the sociological literature and more recently shown in the computer science literature that trust correlates strongly with similarity between people [42, 43]. In our previous work [40] we showed that in addition to overall similarity, there is also a correlation between trust and several nuanced similarity measures. In a context where people rate items, those ratings can be used to compute values that go beyond simple similarity. Specifically, trust between people is tied to the largest single difference over items they have both rated, and to the agreement on movies where one user has given extreme ratings. We also showed that some people tend to be more trusting than others, and thus inferred trust values can be adjusted up or down to account for this. We used these nuanced similarity measures in this research.

#### 3.4.1.3 Experimental Network

When working with trust, data is usually one of the greatest challenges. Trust information is private, and for that reason there are no publicly available datasets with this information. In 2004, we developed and launched FilmTrust,[8] a Web-based social network centered around movies. Users create profiles and link to friends and rate how much they trust each friend's opinion about movies. Users can also rate and review movies. The network has been live on the web and growing on its own since 2004. Thus, it serves as a useful real-world dataset upon which we can run experiments.

The FilmTrust network has 1,610 total members. Many do not have any friends in the social network; 712 people have at least one friend and there are 1,465 edges in the network. The average trust rating is 6.83. The network has a central giant component, and many small subnetworks.

Most users have rated movies in the network; 1,250 people have rated at least one movie. These movie ratings are used in the similarity-based trust inference techniques. In total, we have either trust ratings or movie ratings from 1,339 of the 1,610 users.

### *3.4.2 Experimental Analysis*

Our experimental analysis executes the network-based and similarity-based algorithms over the FilmTrust network, and then follows that with the integrated trust

[8] See http://trust.mindswap.org/FilmTrust.

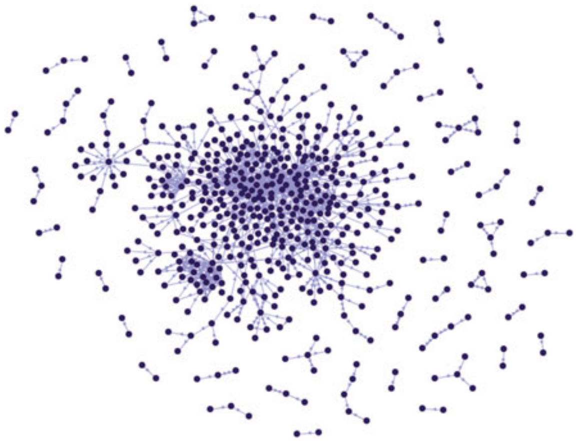


**Fig. 3.11** A visualization of the FilmTrust network

inference algorithm. We compare these methods for accuracy and coverage of the network. We begin this section by describing the setup for each algorithm and then present the results of our experiments. We show that an integrated model does produce significantly more accurate results and better coverage than either method alone (Fig. 3.11).

### 3.4.2.1 Trust Inference Setup

The *network*-based trust inference algorithm, FilmTrust, was able to run directly on the FilmTrust network, so no special setup was required. For the *similarity*-based algorithm, we needed to develop a method for integrating our earlier insights on similarity measures that relate to trust into an algorithm. In that mentioned earlier work we identified four measures made over users' item ratings that relate to trust: overall similarity, similarity on extremes (items that received very high or very low ratings from a user), the single largest difference between users on a given item, and the source's propensity to trust. We computed similarity measures in two ways: as mean average error (MAE) and using the Pearson correlation. Thus, we had six total measures: the average difference (AD), overall correlation (COR), average difference on extremes (XD), correlation on extremes (XCOR), the single largest difference (MD), and the source's propensity to trust (PT). A linear combination of these values can predict trust and is given in Eq. 3.18, where ω indicates the weight given to each measure.

$$\begin{aligned} t_{s,p} = {} & \omega_{\mathrm{AD}} \times \mathrm{AD} + \omega_{\mathrm{COR}} \times \mathrm{COR} + \omega_{\mathrm{XD}} \times \mathrm{XD} \\ & + \omega_{\mathrm{XCOR}} \times \mathrm{XCOR} + \omega_{\mathrm{MD}} \times \mathrm{MD} + \omega_{\mathrm{PT}} \times \mathrm{PT} \end{aligned} \tag{3.18}$$

It is worth noting that for some pairs of people, some of these values may be unavailable. For example, it is common for users to have no movies in common

**Table 3.2** Weights for similarity measures determined in a multivariate linear regression analysis

| Weight | Extreme values available | No extreme values |
|---|---|---|
| $\omega_{AD}$ | −1.8084 | −0.5951 |
| $\omega_{COR}$ | 1.0589 | 0.8269 |
| $\omega_{XD}$ | 0.1751 | |
| $\omega_{XCOR}$ | 0.0655 | |
| $\omega_{MD}$ | 0.2489 | 0.1145 |
| $\omega_{PT}$ | 1.0568 | 0.9946 |

where the source has assigned an extreme rating. Thus, the weights will be different depending on whether or not the two measures on extreme-rated items are available. The weights ($\omega$ values) will vary between networks based on the behavior of the users and context of the data. To determine the optimal weights for the FilmTrust dataset, we ran a multivariate linear regression analysis. To achieve the most meaningful results from the regression, we selected a subset of node pairs who had at least 10 items in common. To compute values for XD and XCOR, we required at least 3 items in common with extreme ratings from the source. The results of this regression analysis are shown in Table 3.2.

#### 3.4.2.2 Integrated Trust Model

Combining the network-based and similarity-based trust inference algorithms into an integrated algorithm has two major benefits. First, it provides a more thorough coverage so trust can be inferred for a greater number of individuals. If someone is not in the social network, the ratings similarity method can be used. If they have not rated enough items but have friends, the social network method can be used. The second benefit is the potential for improved accuracy when trust can be inferred using both the network-based trust inference algorithm and the similarity-based inference algorithm.

Our approach was to use a linear combination of the trust values produced from each base algorithm. To combine these values, we ran a multivariate linear regression analysis using known trust values as a ground truth. We found $\omega_{sim} = 0.869$ and $\omega_{net} = 0.195$. Thus, the integrated trust value was computed as follows:

$$T_{int} = \omega_{sim} * T_{sim} + \omega_{net} \times T_{net} \tag{3.19}$$

### *3.4.3 Results*

The next sections present results on coverage and accuracy of the presented approaches. These experiments were executed on real-world social networks on the Web.

#### 3.4.3.1 Coverage

The FilmTrust network we used for our experiments has 1,610 nodes, but only 1,339 have input any data to the system. Thus, we have $1{,}339 \times 1{,}338 = 1{,}791{,}582$ total pairs for which trust can be inferred. The network has a somewhat high rate of members who have no friends in the social network. Of the 1,339 participating members, only 712 (53.17 %) have any social connections in the network. Recall also that the edges are directed in the network. Nodes must have outgoing edges in order to infer trust to any other nodes. Only 480 nodes have outgoing edges. Thus, we would expect to be able to infer trust values for no more than $480 \times 712 = 341{,}280$ pairs of users if we are using a network-based trust inference algorithm. Note that *any* algorithm that infers trust by searching paths in the network will have this limit. TidalTrust, which infers trust for any sink reachable from the source, was able to compute values for 69,016 pairs. While this is less than 4 % of the total number of pairs, it is just over 20 % of the nodes who have some social network data. The other 80 % of pairs for which a value represent nodes that have no indirect connections in the network (e.g. pairs of nodes not connected to the giant component).

Using the similarity-based method, trust can be inferred for any pair of users who have data in common. In the FilmTrust network, 503,912 ( 28 %) pairs of users have at least one item in common. However, one single item is a very weak basis for computing trust, and is insufficient for computing the correlation measures we need for our method. We set a lower threshold of 3 items in common for computing a similarity-based trust value. With this restriction, 302,336 pairs had an inferred trust value, which is 16.88 % of the total number of pairs

Not surprisingly, when used together, these methods give better coverage than either achieves on its own. We could infer trust for 342,504 pairs, just shy of 20 % of the network. This is less than the sum of the coverage of the two methods, since some pairs have inferred trust from both algorithms. Of the 342,504 pairs for which trust could be inferred, values from *both* methods were available for 28,878 pairs (8.43 % of the covered pairs, and 1.6 % of all pairs).

It is important to note that these coverage rates are unique to the network we are examining here. Other networks may have vastly different coverage rates based on the behavior of the users. Previous work has shown dramatically different rates of participation in the social networking component of websites. However, the improved coverage using two types of algorithms is not surprising and should expected in other datasets (Table 3.3).

**Table 3.3** Pairs of nodes for which trust can be inferred using different methods

| Method | Coverage | (in % of all pairs) |
|---|---|---|
| Similarity-based | 302,366 | (16.88 %) |
| Network-based | 69,016 | (3.85 %) |
| Integrated | 342,504 | (19.12 %) |

#### 3.4.3.2 Accuracy

Improved coverage is useful, but can results from two algorithms also improve the accuracy of trust inferences? We investigated this by using the 1,465 pairs of nodes with a known trust value, i.e. where one user had assigned a trust rating to another in the social network. These were our ground truth values against which the inferred values were compared.

With the TidalTrust algorithm, we tested accuracy by selecting a pair of nodes with a known trust rating, ignoring the edge between them in the network, and then running the algorithm to infer a trust value. This would allow us to see what the algorithm would infer if the relationship did not exist. The similarity-based algorithm was run for any pair of nodes with 3 or more rated items in common.

Of the 1,465 pairs, TidalTrust inferred values for 881 pairs. The similarity-based approach found values for 763 pairs. The intersection of these sets where both methods inferred results comprises 490 pairs. We used those 490 pairs for our analysis so we were comparing the accuracy of all three methods over the same users.

We compared the accuracy of the inferred trust value computed with the integrated trust equation to the accuracy of the trust value inferred using each algorithm individually. Accuracy was measured as both mean absolute error (MAE) and root mean square error (RMSE). For each accuracy measure, an ANOVA indicated statistically significant differences among the three methods. For both accuracy measures, a two-tailed Student's $t$-test showed that the integrated trust method was significantly more accurate than both the network-based and similarity-based trust estimates alone.

This indicates that using both trust inference techniques, the results not only provide inferred trust values for more pairs of users than either method could do alone, but they also can be combined to provide more accurate trust inferences for pairs where both methods generate results.

## 3.5 Discussion and Outlook

In this chapter, we advocated the need for local group trust metrics, presenting our metric Appleseed. Appleseed's nature largely resembles Advogato, bearing similar complexity and attack-resistance properties, but offers one particular feature that makes Appleseed much more suitable for certain applications than Advogato: the ability to compute *rankings* of peers according to their trustworthiness rather than *binary* classifications into trusted and untrusted agents (Table 3.4).

**Table 3.4** Accuracy of trust inference methods

| Method | Accuracy (MAE) | Accuracy (RMSE) |
|---|---|---|
| Similarity-based | 1.36 | 1.78 |
| Network-based | 1.88 | 2.47 |
| Integrated | **1.27** | **1.67** |

Originally designed as an approach to social filtering within our decentralized recommender framework [41], Appleseed suits other application scenarios as well, such as group trust computation in online communities, open rating systems, ad-hoc and peer-to-peer networks.

For instance, Appleseed could support peer-to-peer-based file-sharing systems in reducing the spread of self-replicating inauthentic files by virtue of trust propagation [18]. In that case, explicit trust statements, resulting from direct interaction, would reflect belief in someone's endeavor to provide authentic files.

We also showed that augmenting group trust metrics with additional information that can indicate trust, such as nuanced similarity over rated items, can improve the number of user pairs for whom trust can be inferred.

## References

1. Abdul-Rahman, A., Hailes, S.: A distributed trust model. In: New Security Paradigms Workshop, pp. 48–60. Cumbria, UK (1997)
2. Abdul-Rahman, A., Hailes, S.: Supporting trust in virtual communities. In: Proceedings of the 33rd Hawaii International Conference on System Sciences. Maui, HI, USA (2000)
3. Aberer, K., Despotovic, Z.: Managing trust in a peer-2-peer information system. In: Paques, H., Liu, L., Grossman, D. (eds.) Proceedings of the Tenth International Conference on Information and Knowledge Management, pp. 310–317. ACM Press, Atlanta, GA, USA (2001)
4. Beth, T., Borcherding, M., Klein, B.: Valuation of trust in open networks. In: Proceedings of the 1994 European Symposium on Research in Computer Security, pp. 3–18. Brighton, UK (1994)
5. Ceglowski, M., Coburn, A., Cuadrado, J.: Semantic search of unstructured data using contextual network graphs (2003)
6. Chen, R., Yeager, W.: Poblano: a distributed trust model for peer-to-peer networks. Technical report, Sun Microsystems, Santa Clara, CA, USA (2003)
7. Eschenauer, L., Gligor, V., Baras, J.: On trust establishment in mobile ad-hoc networks. Technical report MS 2002–10, Institute for Systems Research, University of Maryland, MD, USA (2002)
8. Ford, L., Fulkerson, R.: Flows in Networks. Princeton University Press, Princeton (1962)
9. Gans, G., Jarke, M., Kethers, S., Lakemeyer, G.: Modeling the impact of trust and distrust in agent networks. In: Proceedings of the Third International Bi-Conference Workshop on Agent-Oriented Information Systems. Montreal, Canada (2001)
10. Golbeck, J.: Computing and applying trust in Web-based social networks. Ph.D. thesis, University of Maryland, College Park, MD, USA (2005)
11. Golbeck, J.: Combining provenance with trust in social networks for semantic web content filtering. In: Proceedings of the International Provenance and Annotation Workshop, LNCS, vol. 4145, pp. 101–108. Springer, Chicago, IL, USA (2006)
12. Golbeck, J.: Generating predictive movie recommendations from trust in social networks. In: Proceedings of the Fourth International Conference on Trust Management, LNCS, vol. 3986, pp. 93–104. Springer, Pisa, Italy (2006)
13. Golbeck, J., Parsia, B., Hendler, J.: Trust networks on the Semantic Web. In: Proceedings of Cooperative Intelligent Agents. Helsinki, Finland (2003)
14. Guha, R.: Open rating systems. Technical report, Stanford Knowledge Systems Laboratory, Stanford, CA, USA (2003)
15. Guha, R., Kumar, R., Raghavan, P., Tomkins, A.: Propagation of trust and distrust. In: Proceedings of the Thirteenth International World Wide Web Conference. ACM Press, New York, NY, USA (2004)

16. Hartmann, K., Strothotte, T.: A spreading activation approach to text illustration. In: Proceedings of the 2nd International Symposium on Smart Graphics, pp. 39–46. ACM Press, Hawthorne, NY, USA (2002)
17. Jøsang, A., Gray, E., Kinateder, M.: Analysing topologies of transitive trust. In: Proceedings of the Workshop of Formal Aspects of Security and Trust. Pisa, Italy (2003)
18. Kamvar, S., Schlosser, M., Garcia-Molina, H.: The EigenTrust algorithm for reputation management in P2P networks. In: Proceedings of the Twelfth International World Wide Web Conference. Budapest, Hungary (2003)
19. Kinateder, M., Pearson, S.: A privacy-enhanced peer-to-peer reputation system. In: Proceedings of the 4th International Conference on Electronic Commerce and Web Technologies, LNCS, vol. 2378. Springer, Prague, Czech Republic (2003)
20. Kinateder, M., Rothermel, K.: Architecture and algorithms for a distributed reputation system. In: Nixon, P., Terzis, S. (eds.) Proceedings of the First International Conference on Trust Management, LNCS, vol. 2692, pp. 1–16. Springer, Crete, Greece (2003)
21. Król, D.: Propagation phenomenon in complex networks: theory and practice. New Gener. Comput. **32**(3–4), 187–192 (2014)
22. Levien, R.: Attack-resistant trust metrics. Ph.D. thesis, University of California at Berkeley, Berkeley, CA, USA (2004)
23. Levien, R., Aiken, A.: Attack-resistant trust metrics for public key certification. In: Proceedings of the 7th USENIX Security Symposium. San Antonio, TX, USA (1998)
24. Levien, R., Aiken, A.: An attack-resistant, scalable name service. Draft Submission to the Fourth International Conference on Financial Cryptography (2000)
25. Lewicki, R., McAllister, D., Bies, R.: Trust and distrust: new relationships and realities. Acad. Manag. Rev. **23**(12), 438–458 (1998)
26. Marsh, S.: Formalising trust as a computational concept. Ph.D. thesis, Department of Mathematics and Computer Science, University of Stirling, Stirling, UK (1994)
27. Marsh, S.: Optimism and pessimism in trust. In: Ramirez, J. (ed.) Proceedings of the Ibero-American Conference on Artificial Intelligence. McGraw-Hill, Caracas, Venezuela (1994)
28. Maurer, U.: Modelling a public key infrastructure. In: Bertino, E. (ed.) Proceedings of the 1996 European Symposium on Research in Computer Security, LNCS, vol. 1146, pp. 325–350. Springer, Rome, Italy (1996)
29. McKnight, H., Chervany, N.: The meaning of trust. Technical report MISRC 96–04, Management Information Systems Research Center, University of Minnesota, MN, USA (1996)
30. Mui, L., Mohtashemi, M., Halberstadt, A.: A computational model of trust and reputation. In: Proceedings of the 35th Hawaii International Conference on System Sciences, pp. 188–196. Big Island, HI, USA (2002)
31. Page, L., Brin, S., Motwani, R., Winograd, T.: The PageRank citation ranking: bringing order to the Web. Technical report, Stanford Digital Library Technologies Project (1998)
32. Quillian, R.: Semantic memory. In: Minsky, M. (ed.) Semantic Information Processing, pp. 227–270. MIT Press, Boston (1968)
33. Reiter, M., Stubblebine, S.: Path independence for authentication in large-scale systems. In: Proceedings of the ACM Conference on Computer and Communications Security, pp. 57–66. ACM Press, Zürich, Switzerland (1997)
34. Reiter, M., Stubblebine, S.: Toward acceptable metrics of authentication. In: Proceedings of the IEEE Symposium on Security and Privacy, pp. 10–20. IEEE Computer Society Press, Oakland, CA, USA (1997)
35. Richardson, M., Agrawal, R., Domingos, P.: Trust management for the Semantic Web. In: Proceedings of the Second International Semantic Web Conference. Sanibel Island, FL, USA (2003)
36. Sankaralingam, K., Sethumadhavan, S., Browne, J.: Distributed PageRank for P2P systems. In: Proceedings of the Twelfth International Symposium on High Performance Distributed Computing. Seattle, WA, USA (2003)
37. Smith, E., Nolen-Hoeksema, S., Fredrickson, B., Loftus, G.: Atkinson and Hilgards's Introduction to Psychology. Thomson Learning, Boston (2003)

38. Twigg, A., Dimmock, N.: Attack-resistance of computational trust models. In: Proceedings of the Twelfth IEEE International Workshop on Enabling Technologies, pp. 275–280. Linz, Austria (2003)
39. Wiese, R., Eiglsperger, M., Kaufmann, M.: yFiles: Visualization and automatic layout of graphs. In: Proceedings of the 9th International Symposium on Graph Drawing, LNCS, vol. 2265, pp. 453–454. Springer, Heidelberg, Germany (2001)
40. Ziegler, C.N., Golbeck, J.: Investigating interactions of trust and interest similarity. Decis. Support Syst. **43**(2), 460–475 (2007)
41. Ziegler, C.N., Lausen, G.: Paradigms for decentralized social filtering exploiting trust network structure. In: Meersman, R., Tari, Z. (eds.) Proceedings of the DOA/CoopIS/ODBASE Confederated International Conferences (2), LNCS, vol. 3291, pp. 840–858. Springer, Larnaca, Cyprus (2004)
42. Ziegler, C.N., Lausen, G.: Spreading activation models for trust propagation. In: Proceedings of the IEEE International Conference on e-Technology, e-Commerce, and e-Service. IEEE Computer Society Press, Taipei, Taiwan (2004)
43. Ziegler, C.N., Lausen, G.: Propagation models for trust and distrust in social networks. Inf. Syst. Front. **7**(4–5), 337–358 (2005)

# Chapter 4
# Assessing the Role of Network Effects in Propagation Phenomena in Real World Networks

**Newton Paulo Bueno**

**Abstract** Despite the growing evidence that interpersonal and social dynamics play a major role in the formation of beliefs, feelings and behaviors, studies on dynamic systems seldom mention the role of social networks in shaping MMs and thus in affecting systems behavior. The general purpose of this chapter is to contribute to bridge that gap by presenting a number of structural properties of social networks that can influence the propagation of ideas, beliefs and behaviors that shape mental models. Specifically it aims at three goals: (a) to show how to identify different types of social networks, (b) to discuss how these different structures can either promote or hinder the adaptation of mental models in problematic systems, and (c) to discuss some basic strategies to fix inadequate mental models in different types of networks.

## 4.1 Introduction

According recent studies social interaction is of paramount importance for the spread of behaviors and beliefs ranging from obesity to willingness of women in less developed countries to adopt contraceptive measures [7, 10, 26]. How fast these different types of influences travel through social networks shaping people's mental models, as it's been shown, depends on certain key properties of their structure. For example, studies in diffusion have found that centralization and radiality (terms to be defined below) are positively related to innovativeness in social networks [3], whereas assortative biases in the joint distribution of attributes such as the number of relations that agents have within the network, race, age and social status influence negatively diffusion [18, 23].

Yet, if the problem were only that the speed of diffusion may vary with the network structure, using traditional differential equation approaches as in epidemiology for studying propagation of social phenomena might be justifiable [20, 25]. The real problem arises when social networks display certain structural properties—such as high centralization and disassortativity which indicate that network effects are more

N.P. Bueno (✉)
UFV-Federal University of Vicosa, Vicosa, Minas Gerais, Brazil
e-mail: npbueno@ufv.br

D. Król et al. (eds.), *Propagation Phenomena in Real World Networks*,
Intelligent Systems Reference Library 85, DOI 10.1007/978-3-319-15916-4_4

important than system (feedback) effects in shaping mental models. Under those circumstances, generalized statements about, say, policy effectiveness across different policy domains lack validity, because even small changes in parameters can be magnified dramatically by network effects. SIR-type system dynamics models (sd), for instance, can of course display tipping point properties due to what we may call systemic effects, i.e. changes in feedback loop dominance [5, 24]. We are not aware, however, of any procedure to assess the impact of network effects in actual systems only by means of traditional SD diffusion models, which are based on homogenous-mixing assumptions.[1] This suggests that other computational approaches, such as agent based modeling and network analysis, are called for to be used in conjunction with SD models to fully address the impact of network properties on diffusion of ideas that can shape MMs [11, 26, 29].

Yet, there are a number of structural properties of social networks that can qualitatively affect diffusion paths of ideas, beliefs and feeling, and hence mental models formation, in relatively predictable ways. As decision makers do not usually need to develop perfect and complete mental models of complex environments to reach better decisions but only understand key principles, uncover those qualitative patterns may suffice to improve performance in most of dynamic environments [13].

The purpose of this chapter is to discuss how basic network properties can affect the diffusion of ideas through social networks, showing how these different structures can either promote or hinder the adaptation of MMs, and indicating what strategies may be used for improving MMs in problematic systems (see footnote 1).

## 4.2 Basic Concepts and Ideas on Networks

Different types of social networks have different percolation thresholds and hence present different diffusion patterns. The question of percolation is equivalent to whether there exists a path from one side of the network to the other through which information can flow. When there is such a path we define the giant component as the unique largest connected component of a network. The presence of a giant component implies that a macroscopic fraction of the network is connected, which plays a central role in diffusion problems. For instance, the size of a particular giant component gives an idea of the most vertices that one might possibly reach starting from a single vertex [16].

The determination of whether a network presents a giant component large enough to allow rapid diffusion of information depends ultimately on its centralization degree relatively to its average degree. An actor's degree is the number of relations he has within the network. A network's mean degree is the average number of relations

[1] For example, the Bass model that assumes that interpersonal influence is an essential component in the diffusion process is not, strictly speaking, a network model. It avoids discussing contact networks at all by making use of a fully mixed approximation, in which it is assumed that every individual has an equal chance of coming into contact with every other, per unity of time.

between all vertices of the network. The centrality of an entire network indexes the tendency of a single vertex to be more central that all other points in the network [12].

A network is said to be assortatively mixed if agents (vertices) are more likely to maintain relationships with whom are similar to themselves in patterns attributes such as centrality degree, race, age, social status. In diassortative networks, in contrast, high degree vertices such as the opinion leaders in social networks—tend to be connected to low degree ones, which are relatively disconnected from each other. A network is said to be highly centralized if one vertex or a small number of vertices have much higher degree than the other vertices (or if it is easily accessible all other unities or if it lies on several shortest paths between other unities). Relatively higher variance to the mean degree, which indicates a larger centralization degree, makes a network more conducive to information diffusion, as it provides higher degree nodes to lead to the formation of giant components.

The critical threshold of percolation, that is the mass of adopters of new attitudes or behaviors required to trigger the formation of a giant component, depends on two moments of networks degree distribution: the mean degree (4.1) and the squared mean degree (4.2).

$$\langle k \rangle = \sum_{k=0}^{\infty} k p_k \tag{4.1}$$

$$\langle k^2 \rangle = \sum_{k=0}^{\infty} k^2 p_k \tag{4.2}$$

The critical percolation threshold is given by [20]

$$\phi C = \frac{\langle k \rangle}{\langle k^2 \rangle - \langle k \rangle} \tag{4.3}$$

The size of the corresponding giant component S, that is the fraction of agents that are in the giant component (the formula is valid for a random graph but can be used to approximate the size of giant cluster is other types of networks as well) can be computed numerically through the following equation [20]:

$$S = 1 - e^{-\langle k \rangle S} \tag{4.4}$$

Which can be rewritten as in Eq. 4.5:

$$k = \frac{\ln(1 - S)}{-(S)} \tag{4.5}$$

Thus, if for instance the mean degree is greater than five, nearly all agents will become members of the giant cluster. If it is lower or equal than 1, on the other hand, a giant cluster cannot emerge. For checking this result, notice that the giant cluster consistent with a mean degree of 1.025 would contain only 5 % of agents, while the giant cluster that could emerge in a network with a mean degree of 1.0025 would contain less than 1 % of the agents. For mean degrees lower than 1, there are no consistent positive values for S.

The main metrics that can be used to measure network characteristics that influence the level of resilience or flexibility of mental models are as follows.

### *4.2.1 Mean Degree*

The mean degree of a network—k—is positively connected to diffusion. The explanation is that, according Eqs. (4.3 and 4.4), the higher the mean degree of a social network the sooner a new idea or belief will percolate and the larger the size of the giant cluster that will emerge. In low connected networks—mean degree lower than 1—the separation between vertices could be large, and there is no giant cluster, instead the network consists of many components that are small relative to the number of nodes. For more connected and relatively homogeneous in-degree networks, on the other hand, we can compute their epidemic threshold (that is the line that separates an initially growing number of people adopting a new idea from an initially decreasing one) exactly like we do in system dynamics SIR-type models. Specifically, we can easily show that the lower the mean degree of a network, the higher must be the probability of infection relatively to the recovery rate in order to new ideas to spread [20].

### *4.2.2 Centralization Degree*

A more centralized network speeds up diffusion because the critical mass for percolation is reached sooner than in less centralized ones. For confirming this result, observe that the larger the heterogeneity in degree and therefore the centralization of a network, the larger the denominator of the formula for computing its percolation threshold (Eq. 4.3). Thus, new ideas and behaviors can propagate rapidly once they reach some high ranking nodes, i.e. the hubs [22]. Most of complex highly centralized systems, such as the WEB, the cell, and scientist and professional networks, contain hubs. An important result from recent studies in network analysis is that highly centralized networks are less vulnerable to random failures but more fragile to targeted attacks to high ranking nodes. Lower centralized homogeneous networks, on the other hand, are more fragile to random shocks, once they are close enough their percolation threshold [2]. This result may have important implications for the speed with which people adapt their mental models to new information, as we shall discuss later.

### 4.2.3 Assortativity Degree

Assortativity correlates negatively to diffusion due to the homophily principle, that is due to the fact that people are more likely to maintain relationships with whom are similar to themselves. Assortativity can be associated with the emergence of a core-periphery structure, in which a set of a densely connected actors constitute the core of the network while many other (low degree) actors constitute the periphery.

More connected people in assortatively mixed networks (e.g., agents 1 and 2) link mostly to each other, whereas in disassortatively mixed networks there are highly connected actors (the opinion leaders) who connect relatively disconnected people to the social network's core.

Figures 4.1 and 4.2 depict a computer generated assortatively mixed network at the top and a disassortative one at the bottom. As new ideas usually enter a system through higher status members (e.g., vertices 1, 2), a higher degree of homophily would mean that new information usually would not reach the periphery in assortatively mixed networks. It would tend to spread horizontally, rather than vertically, which slows down the rate of diffusion. Heterophilous disassortatively mixed networks, in contrast, are said to be radial ones which aid rapid diffusion [28]. In this class of radial networks followers (e.g., vertices 12, 13, 14) tend to seek the more connected opinion leaders of higher socioeconomic status and/or more formal education, perceived as more technically competent (e.g., vertices, 1, 11). This makes the most influential opinion leaders potential key targets in diffusion campaigns because they are in general influential in political, health, agricultural and educational issues [23].

One way to assess the assortativity degree of a social network is by computing an assortativity coefficient (r), based on the correlation between centrality degrees. A positive value for r indicates a tendency to high degree nodes to connect to other high degree nodes.

Table 4.1 displays the calculated r for some assortatively and disassortatively mixed systems, which confirms the insight that there is a tendency for the social networks to have a positive r, indicating assortatively mixing by degree, while technological and biological networks have negative links, indicating disassortatively mixing [21].

An alternative way to access the assortativity degree of a social network of interest is by testing whether a given dataset has a hypothesized core-periphery structure as in the adjacency matrix in Fig. 4.2 [3]. A built-in function of the computer program UCINET, to be used in Sect. 4.4, allows us to compute the Pearson correlation coefficient (fitness) between actual data sets and the adjacency matrix in Fig. 4.2.

**Table 4.1** Correlation in degrees

| | Math coautorship | Film actors | Power grid | Internet | Train routes | Marine food web |
|---|---|---|---|---|---|---|
| r | 0.120 | 0.208 | −0.003 | −0.0189 | −0.0333 | −0.263 |

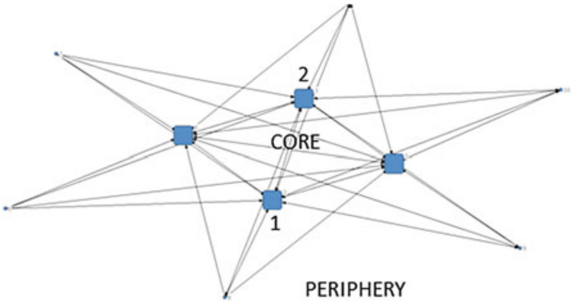


**Fig. 4.1** Assortatively mixed network

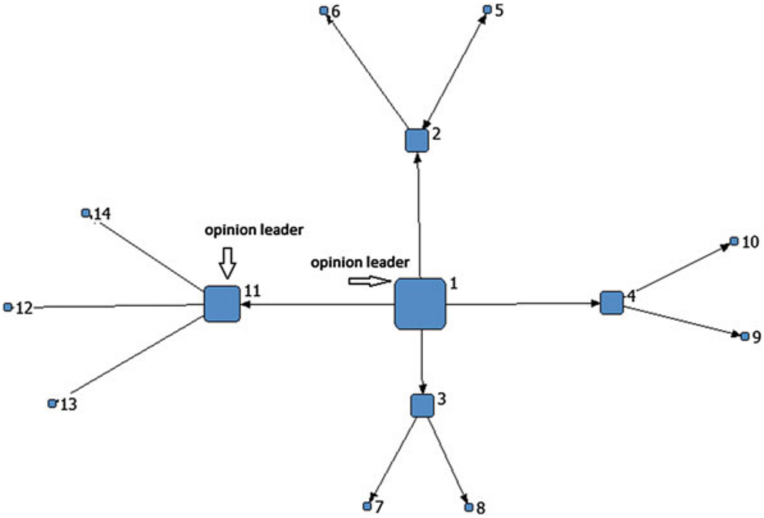


**Fig. 4.2** Disassortatively mixed network

## 4.3 Basic Types of Social Networks

Recent research in network science has identified three types of networks as of more interest to applied human sciences. Random networks in which links are randomly distributed across vertices and form a bell-shaped distribution; in this kind of network, most of vertices have a typical number of links, with the frequency of remaining rapidly decreasing on either side of the maximum. Most of real social networks, however, follow the so-called small world pattern according which communities organize along homophilous links, presenting weakly connected clusters of individuals, who have relatively small numbers of connection outside their own clusters, and small path length between apparently widely separated vertices [30]. One example of this type of network is the one studied by Granovetter [15] which showed that we get most relevant information from acquaintances linked to us through weak and long range ties, and not from close friends. A third type is defined as scale free networks, where most vertices have only a few links, and a tiny minority the hubs are disproportionately highly connected, such as the World Wide Web.

The latter, therefore, present skewness in their degree distribution, which is the typical signature of social network in which network effects have an important influence on how agents behave. The highly connected agents in the network may exercise a powerful influence on the behavior of the low connected ones [22]. If they can be persuaded to change their beliefs or behaviors this can trigger a cascade of behavior change though the entire social network, as predicted by the two-step flow model of influence [17, 32]. In small-world networks in which hubs are not supposed to exist, on the other hand, attempts to identify and influence opinion leaders are a waste of time and resources.

A better strategy to encourage the spread of new ideas or behaviors through these systems is to attempt to influence a critical mass, for instance by advertisement like in the Bass model, which would allow the news to percolate across the population by word-of-mouth.

Network science, thus, explains us why some campaigns seeking changing long lasting mental models in fields such health, politics and sustainable practices and technologies fail to spread while other have a dramatic impact, namely the fact that the effectiveness of a policy will be contingent on the type of network upon which it is being enacted. If a particular network is best approximated by a random or a small-world pattern, then the concept of influential is not so important, and qualitative paths of ideas and behaviors diffusion can be roughly predicted through traditional system dynamics models of diffusion. But if the networks resemble a scale-free pattern, conclusions based on those models can be severely misleading, which suggests the need of complement analysis with other approaches such as agent based modeling.

## 4.4 Illustration: Assessing the Network Structure of a Small Irrigation System

In order to illustrate the above reasoning we created a NetLogo landscape for an actual irrigation project the Gorotuba River irrigation district—in the northwest region of the state of Minas Gerais/Brazil [33].

The landscape, depicted in Figs. 4.3 and 4.4, was populated with 450 families and 40 firms. The thicker line represents the main irrigation channel while the thinner ones represent the secondary channels of the project. Families and firms were placed in their actual locations in the district. We assumed that firms differ from families in that they are less lazy statistician than families, in the sense they search around by a larger radius to get information about the environment before taking actions.

The number of the agents located within the searching radius, measured for the geodesic distance among cells in the grid, was used as a proxy to each agent's centrality degree in their social network. Based on these assumptions we built a binary social matrix in which each cell assumes value zero if agents are not related to each other and 1 otherwise. We finally built the graph for the social network of

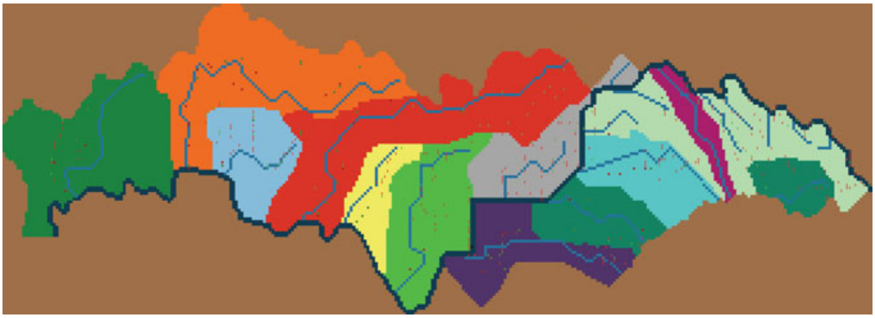


**Fig. 4.3** The Gorutuba River irrigation district: wide view

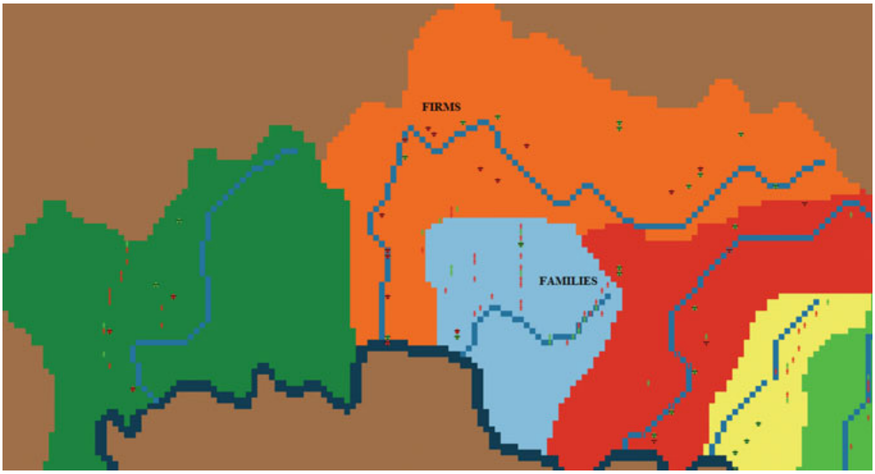


**Fig. 4.4** The Gorutuba River irrigation district: partial view

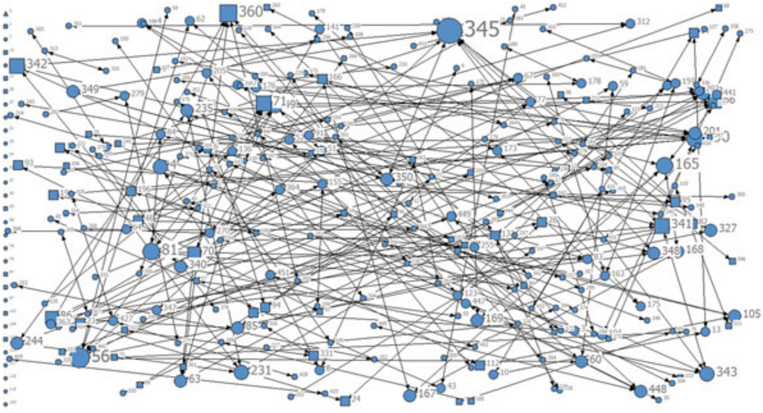


**Fig. 4.5** Social network of the irrigation district of Gorotuba River: centrality degree

**Table 4.2** Structural properties of the Gorotuba River irrigation district's underlying social network

| Rewiring level | Mean degree | r | Centralization (%) | Percolation threshold | Fitness to an idealized core-periphery structure |
|---|---|---|---|---|---|
| 0 | 0.96 | 0.272 | 1.54 | 0.82 | 0.18 |
| 1 % | 9.87 | 0.008 | 2.22 | 0.07 | 0.07 |

the Gorotuba irrigation district in Fig. 4.5 with basis on the social matrix above with the computer program UCINET [4].

The next step was randomly rewiring the social network in order to allow agents to establish relations with more distant neighbors. Hence a 1 % of matrix rewiring for instance means that agents link now to previously disconnected agents with 1 % of probability, that is that each agent establishes connection with four new neighbors, on average. As expected from previous studies in social networks (see for instance [30], even a few extra links are sufficient to drastically increase the average connectivity among agents, as measured by the network's mean degree (Fig. 4.6).

Are rewired networks more conducive to the diffusion of new ideas than the original network? Table 4.2 displays some key structural network's properties calculated as explained in Sect. 4.2 for two different rewiring levels of the social network at hand.

While the difference in centralization is insignificant, there seems to be a tendency for the non-rewired network to display a structure less favorable to information propagation. For instance, the mean degree of 0.96 implies that a giant component of adopters is unlikely to emerge in the non-rewired network [16] whereas the percolation threshold of 0.82 means that more than 80 % of the population has to be infected

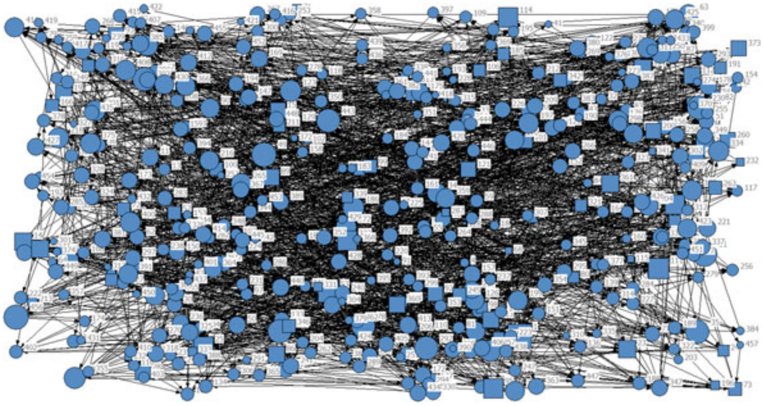


**Fig. 4.6** Social network of the irrigation district of Gorotuba River: centrality degree (1 % rewired matrix)

in order to trigger a cascade of information. The rewired network is also more radial, i.e., less assortative than the actual one.

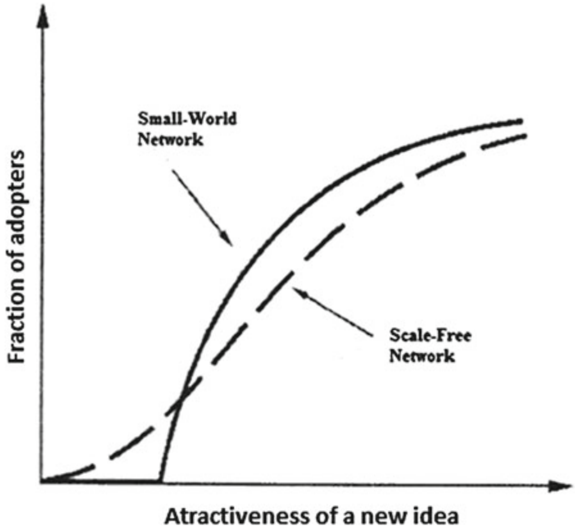
Assortatively mixed networks usually display structural characteristics of small world networks, such as relatively high percolation thresholds, which means that mental models in such settings can remain stable and resilient to new information for a long time, until, unexpectedly, attitudes and behavior change in a cascade in response to seemingly unimportant changes, for example in environmental parameters as a reduction in the amount of rainfall.

The problem for the diffusion of information in more homogeneous and less connected networks like the Gorotuba district, specifically, is that the percolation threshold is high because producers are too weakly connected to trigger cascades of adoption of, say, more sustainable behaviors through their underlying social networks.

This seems to be a problem even in developed countries in which farmers are supposed to be more informed and conscious on the need of adopting sustainable irrigation practices. In Australia, for example, while the level of uptake of drought prediction is high within government agencies, less than one third of farmers take drought predictions into account. Furthermore, farmer's preparedness to change major decisions is not generally influenced by this information. This is true even in extreme situations such as El Niño events, and despite the fact that drought link is widely known and accepted within the Australian agricultural community [34].

The rewired Gorotuba network, on the other hand, tends to present some characteristics of scale free networks, such as relatively highly connected hubs, capable of disseminating information widely. These networks differ from small worlds in that social contagious does not require any epidemic threshold is crossed. The explanation is that, in the presence of highly connected hubs, social contagion may spread

**Fig. 4.7** The spread of new ideas in small-word and scale-free network



quickly if the disease infects even a single hub near a vulnerable cluster of agents (for examples of this type of scenario, see [19]). Figure 4.7 (Adapted from [32]) compares typical diffusion paths in assortative and disassortative networks.

The attractiveness of a new idea is given for the probability of an agent to be convinced to adopt a new idea by a former adopter. The larger the contact rate (k), that is the mean degree of the network, the larger the number of people exposed to the new idea, the faster the spread of the new idea throughout the network (the smaller the percolation threshold in Eq. 4.3 and the larger the size of the cluster of final adopters (Eq. 4.5).

## 4.5 Discussion and Conclusion

The process of adaptation of mental models to new information tends to follow different qualitative patterns in different kinds of social networks. Networks in which social processes can play the major role, such as the spread of infectious diseases, rumors, riots and fads tend to be characterized by assortative mixing in-degree as it is typical of small-world networks [31]. In contrast, socio-ecological networks and large size techno-social systems, such as transportation systems and power distribution grids, are in general characterized by disassortative mixing in-degree and hence tend to resemble scale-free networks [29]. Economic networks, finally, present properties of both disassortatively and assortartively mixed systems [16].

In the most common type of purely social networks—small-worlds—as there are no especially influential agents, a relatively large critical mass of infected individuals is required for triggering mental model changes. In technological and economic networks, in contrast, if a small number of highly connected agents are influenced by new ideas and beliefs the whole process can be very rapid. Our simulations, however, suggest that the addition of a few extra links among agents into small-world networks can dramatically reduce their percolation threshold i.e., the size of the critical mass of infected individuals required to trigger a cascade of change and, hence, decrease the length of the time delay required to adjust mental models.

These results indicate that there is a considerable room for policies seeking to improve mental models deemed inadequate, though the kinds of approaches which might work on a type of network may have little impact on another. We can think of three basic types of strategies for doing so: (a) to identify opinion leaders and convince them to change their mental models; (b) to try to percolate clusters of individuals with new ideas and behaviors and (c) to alter the structure of the network.

The first strategy probably works best in technological systems, such as the internet and the world wide web, which are ultimate scale-free disassortatively mixed networks As such networks are robust against random failures but vulnerable to attacks on their highest degree vertices, the easier way to encourage mental models changes is to persuade such highly connected vertices, say the more connected persons on Facebook. These persons, even when they are otherwise ordinary people, are important simply because they have so many connections. When diffusion starts with these individuals the delay time between the introduction of an innovation and the emergence of the critical mass required to trigger its diffusion is eliminated [27].

That can be true with the proviso that is many times easier said than done because it can be hard to influence these people. Studies in diffusion have shown that opinion leaders have followers but in general are not willing to be the first to adopt new ideas [23].

In small-world-like networks, on the other hand, elite individuals interact mainly with one another which implicates that the first strategy is often ineffective. If we nevertheless attempt to introduce changes in mental models through higher status individuals, the new ideas would tend to spread horizontally not trickling down to non-elites.

Yet, as these systems generally exhibit tipping point properties, it is theoretical possible to speed up MM changes if we manage to percolate clusters of similar people by persuading a critical number of them. An even more efficient strategy, inspired in vaccination campaigns, might be to attempt to influence the friends of randomly picked individuals. The non-intuitive rationale for this strategy is that for the most types of social networks, the agents connected to any set of randomly selected ones tend to be more connected than they own [6].

More specifically, a random sample of individuals will have a mean degree of $\mu$ (the mean degree for the population); but the friends of these random individuals will have a mean degree de plus a quantity defined by the variance of the degree distribution divided by $\mu$ [8].

The third strategy to encourage changes in mental models is to change the connection between people rather than focusing on getting people individually to change their behavior. For example, if a network is much sparse, like in the non-rewired version of the Gorotuba irrigation district, the propagation of information across it is likely to be very limited (because a giant cluster of even modest size is unlikely to emerge if the network's mean degree is low). Policies seeking to change MMs in this case should be directed toward increasing social connections among agents.

Another example is on measures capable to prevent the spread of undesirable behaviors. For instance, there is currently clear evidence that obesity is a social phenomenon, in the sense its prevalence occurs through social networks with clear boundaries. That is because the social pressure and influences on not to become obese are relaxed when other people are already obese. Accordingly, Christakis and Fowler [7] have identified discernible clusters of obese persons, which are not attributable to the selective formation of social ties among obese persons, but to the fact that people are influenced the most by whom they resemble, which means that homophily plays a key role in these networks formation. This and other public health issues, such as alcoholism and drug addiction, therefore, seems to require interventions based on measures that provide peer support, that is that modify the person's social network to another in which prevalent mental models do not see obesity, for example, as the default mode of the underlying social network.

Just to give a very preliminary idea on how to use network concepts to analyze a real-world situation, consider the mass protests that have taken place recently in Brazil. Since mid-2013, the country has experienced waves of increasingly violent riots against corruption and the low quality of public services, which politicians and intellectuals have proved completely unprepared to understand because mental models regarding the problem seem, at least partially, wrong.

The mental model often made explicit by members of the federal government and by some influent intellectuals, in the first place, is the protests even the violent ones are essentially legitimate, which means that measures to reduce political violence should be more political than punitive. The President herself, a politician with a leftist orientation, received leaders of one these events at office just one day after they had wounded police officers and attempted to violently invade the president's office. Protesters' mental model, secondly, which has certainly been encouraged by government behavior, appears to be that the cost/benefit of adopting increasingly violent attitudes is favorable as that behavior allows the most violent among them to assume a position of prominence in their social networks. Public opinion finally, after having enthusiastically supported a first wave of (peaceful) protests, began to see them more and more as manifestations of troublemakers that need to be suppressed. The murder of a journalist who was following one of these events by rioters in Rio de Janeiro apparently have contributed to spread this sense of outrage among the majority of the population and part of the authorities. But what exactly should be done?

Two main strategies, thus far only vaguely outlined, are on the table. The first one, sponsored by influent members of the government and by most of intellectuals, is to try to identify and convince protest leaders to change their behavior by means of negotiation around issues like bus and metro ticket prices, which might induce, at least in theory, their followers to do the same.

The second strategy championed basically by police members and by some politicians from more conservative opposition parties, is to make the laws more severe, ranking violent acts in protests as terrorism, which in principle would increase the costs of violent action for perpetrators. This is not the place to try to assess the merits of each strategy. Yet, our analysis gives some (speculative) clues about the possible consequences of each one.

It suggests, for instance, that a prerequisite for the first strategy work is that the protesters' network is disassortative, i.e., that there are influentials to be influenced. Moreover, this measure would make sense only if the supposed leaders could in fact be induced to change their behavior, which seems at best unlikely, since these people would have to abandon just the practices that have brought them notoriety among their peers. Secondly, it informs us that a condition for the second measure to work is that the protesters' network is predominantly assortative. Only in this kind of network, it would make sense to attempt inducing cascades of changes in mental models by percolating clusters of relatively homogenous protesters using the fear of more rigorous punishments. Even so, it would be necessary that the police forces and justice courts were sufficiently efficient to identify and remove a critical mass of troublemakers from the network, which is far from being certain at this point.

While the insights and simulations presented in the chapter shall be seen more like thought experiments than real experiments, we believe they can have important implications for both empirical and theoretical studies in system dynamics. The most important of them seems to be we can no longer think of people as only individuals reaching carefully considered decisions as social networks foster strong norms about a wide range of topics, including life goals, moral values, and even clothing. People's desires and preferences, thus, are mostly based on what respective peer community agrees is valuable rather than on rational reflection based directly on their individual biological drives or inborn morals. There is now robust evidence that, in these circumstances, providing social network incentives to change mental models is a far more powerful method of changing behaviors than the traditional method of using individual incentives.

Models of social interaction have been developed to explain the observation of large differences in outcomes in the absence of significant differences in fundamentals. In those models each persons action changes not only in response to direct changes in fundamentals, but also because of the change in the behavior of his or her peers. These models show that if social interaction is large enough one may observe different outcomes from exactly the same fundamentals.

In a classic study in the field, Glaeser et al. [14] showed that the large variation in crime rates across large American cities at that time could be better explained by a model of social interaction rather than by the usual social-economic variables. More recently, a number of economic studies have broadened this framework to analyze, for instance, the occurrence of large drops in aggregate economic activity due to the propagation of microeconomic shocks through input-output linkages across different firms or sectors within the economy [1], interbank risk contagion caused by idiosyncratic small shocks [21], and the adoption of new products. For instance, commercials in big events are often used to advertise products where network effects are important in the sense it is rational to be one of a large population of adopters. A classic example was the introduction of the Apple Macintosh in the 1984 Super Bowl by a commercial directed by Ridley Scott in which Apple did not simply inform each viewer about the product, but also told each viewer that many other viewers were informed about the Macintosh [9]. Overall, therefore, as the patterns of connections in a network can have profound impacts on the propagation of ideas, beliefs and behaviors in actual settings, it seems important to develop a better understanding of characteristics that describe who interacts to whom in actual settings.

**Acknowledgments** CNPq and FAPEMIG are gratefully acknowledged for their financial support.

## References

1. Acemoglu, D., Ozdaglar, A., Tahbaz-Salehi, A.: The network origins of large economic downturns. Technical report. National Bureau of Economic Research (2013)
2. Barabási, A.L.: Linked: The New Science of Networks. Basic Books, New York (2002)
3. Borgatti, S.P., Everett, M.G.: Models of core/periphery structures. Soc. Netw. **21**(4), 375–395 (2000)
4. Borgatti, S.P., Everett, M.G., Freeman, L.C.: UCINET for windows: software for social network analysis (2002)

5. Bueno, N.P.: Assessing the resilience of small socio-ecological systems based on the dominant polarity of their feedback structure. Syst. Dyn. Rev. **28**(4), 351–360 (2012)
6. Cho, A.: Ourselves and our interactions: the ultimate physics problem? Science **325**(5939), 406–408 (2009)
7. Christakis, N.A., Fowler, J.H.: The spread of obesity in a large social network over 32 years. N. Engl. J. Med. **357**(4), 370–379 (2007)
8. Christakis, N.A., Fowler, J.H.: Social network sensors for early detection of contagious outbreaks. PloS One **5**(9), e12,948 (2010)
9. Chwe, M.S.Y.: Rational ritual: culture, coordination, and common knowledge. Princeton University Press, Princeton (2001)
10. Coleman, J.S., Katz, E., Menzel, H., of Applied Social Research, C.U.B.: Medical Innovation: a Diffusion Study. Bobbs-Merrill Co., Indianapolis (1966)
11. Eubank, S., Guclu, H., Kumar, V., Marathe, M., Srinivasan, A., Toroczkai, Z., Wang, N.: Modelling disease outbreaks in realistic urban social networks. Nature **429**(6988), 180–184 (2004)
12. Freeman, L.C.: Centrality in social networks conceptual clarification. Soc. Netw. **1**(3), 215–239 (1978)
13. Gary, M.S., Wood, R.E.: Mental models, decision rules, and performance heterogeneity. Strateg. Manag. J. **32**(6), 569–594 (2011)
14. Glaeser, E.L., Sacerdote, B., Scheinkman, J.A.: Crime and social interactions. Q. J. Econ. **111**(2), 507–548 (1996)
15. Granovetter, M.S.: The strength of weak ties. Am. J. Sociol. **78**(6), 1360–1380 (1973)
16. Jackson, M.O.: Social and Economic Networks. Princeton University Press, Princeton (2008)
17. Katz, E., Lazarsfeld, P.: Personal influence: the part played by people in the flow of mass communication, Transaction Publishers, Livingston (2005)
18. Morris, M.: Local rules and global properties: modeling the emergence of network structure. In: Dynamic Social Network Modeling and Analysis, pp. 174–186. National Academies Press, New York (2003)
19. Motter, A.E.: Cascade control and defense in complex networks. Phys. Rev. Lett. **93**, 098,701 (2004)
20. Newman, M.: Networks, an Introduction. Oxford University Press, Oxford (2012)
21. Nier, E., Yang, J., Yorulmazer, T., Alentorn, A.: Network models and financial stability. J. Econ. Dyn. Control **31**(6), 2033–2060 (2007)
22. Padgett, J.F., Ansell, C.K.: Robust action and the rise of the medici, 1400-1434. Am. J. Sociol. **98**(6), 1259–1319 (1993)
23. Rogers, E.M.: Diffusion of Innovations. Simon and Schuster, New York (2010)
24. Rudolph, J.W., Repenning, N.R.: Disaster dynamics: understanding the role of interruptions and stress in organizational collapse. Adm. Sci. Q. **47**(1), 1–30 (2002)
25. Tutzauer, F., Knon, K., Elbirt, B.: Network diffusion of two competing ideas. In: The Diffusion of Innovations a Communication Science Perspective, pp. 145–170. Peter Lang, New York (2011)
26. Valente, T.W.: Network models and methods for studying the diffusion of innovations. Models Methods Soc. Netw. Anal. **28**, 98 (2005)
27. Valente, T.W., Davis, R.L.: Accelerating the diffusion of innovations using opinion leaders. Ann. Am. Acad. Polit. Soc. Sci. **566**, 55–67 (1999)
28. Valente, T.W., Foreman, R.K.: Integration and radiality: measuring the extent of an individual's connectedness and reachability in a network. Soc. Netw. **20**(1), 89–105 (1998)
29. Vespignani, A., et al.: Predicting the behavior of techno-social systems. Science **325**(5939), 425–428 (2009)
30. Watts, D., Strogatz, S.: Collective dynamics of small-world networks. Nature **393**, 440–442 (1998)
31. Watts, D.J.: The "new" science of networks. Annu. Rev. Sociol. **30**, 243–270 (2004)
32. Watts, D.J., Dodds, P.S.: Influentials, networks, and public opinion formation. J. Consum. Res. **34**(4), 441–458 (2007)

33. Wilensky, U.: NetLogo. http://ccl.northwestern.edu/netlogo/
34. Wright, W.: Significance of training, education and communication for awareness of potential hazards in managing natural disaster in australia. In: Sivakumar, M., Motha, R., Das, H. (eds.) Natural Disasters and Extreme Events in Agriculture, pp. 219–239. Springer, Berlin (2005)

# Chapter 5
# Resource Constrained Randomized Coverage Strategies for Unstructured Networks

**Subrata Nandi and Niloy Ganguly**

**Abstract** Most of the information management algorithms in large-scale unstructured networks, require one to maximize the coverage ($C$) i.e. expected number of distinct visited nodes. An unstructured system lacks an index. Hence, to maximize coverage, flooding-based strategies are used when there is an abundance of bandwidth ($B$) and single random walker is used in abundance of time ($T$). However, there exists an inherent tradeoff between coverage speed and bandwidth utilization. Hence, in practical scenarios, when the amount of resource, both bandwidth $B$ and time $T$ are finite, it is nontrivial to design strategies that maximizes network coverage. This chapter defines the extended network coverage problem under constrained resources, reveals the underlying challenge and discusses some of the recent works that design randomized strategies to maximize coverage $C(B, T)$. Specifically, the chapter explains how the understanding of $K$-random walk dynamics has been used to develop uniform and non-uniform proliferating random walk strategies to achieve the goal. These coverage strategies may be useful in designing efficient services, e.g. search, gathering and routing for large scale networks like sensors, peer-to-peer, computing grids, etc.

## 5.1 Introduction

Information management [1] in large-scale distributed systems ranging from social networks and the Internet to ad-hoc [17], peer-to-peer (P2P) [7, 14, 23], delay tolerant networks [12], sensor [16] and mesh networks, computing grid etc. require services like search [1, 4, 6, 10], dissemination [30, 33], gathering [24], spreading [18, 19] and routing [35, 40] of information. Depending on how the data is organized, such networks may be classified into two broad categories—structured and

S. Nandi (✉)
National Institute of Technology, Durgapur 713209, India
e-mail: subrata.nandi@gmail.com

N. Ganguly
Indian Institute of Technology, Kharagpur 721302, India
e-mail: niloy@cse.iitkgp.ernet.in

D. Król et al. (eds.), *Propagation Phenomena in Real World Networks*,
Intelligent Systems Reference Library 85, DOI 10.1007/978-3-319-15916-4_5

unstructured. In structured networks an index or hash table is maintained whereby the index is used to locate the node containing the data, whereas in unstructured networks, no such indices are stored explicitly. As a result, unstructured networks do not incur any overhead for maintaining the index structure and can handle high churn rates efficiently. Large networks are inherently dynamic and follow a decentralized architecture as they lack centralized control or authority. Hence, information in such systems is generally organized in an unstructured fashion.

All real-world networks work under resource constraints. Assume for its execution, each service request be allocated $\mathcal{T}$ time steps and $\mathcal{B}$ units of bandwidth quota, as resource. **Bandwidth** here refers to the total number of messages required to be passed among the nodes corresponding to a service request. Hence, it is an implicit measure of physical resource, the amount of communication bandwidth and battery power used by a service. **Time** refers to the maximum latency that is allowed for the execution of a service starting from its initiation. *Starting from a single node, most of the underlying information management algorithms for unstructured networks, require to maximize the expected number of distinct visited nodes, the* **coverage** $[C(\mathcal{B}, \mathcal{T})]$ *of the network, consuming $\mathcal{B}$ amount of bandwidth within a service time of $\mathcal{T}$ units.*

It may be intuitively noted that by fixing the maximum lifetime $\mathcal{T}$ of a service, real network systems also place an implicit limit to the overall amount of bandwidth consumption per service request, thereby resource constraint. However, to know the underlying challenge and classical flavor in designing such resource constrained coverage strategies, one require to get an insight of how the optimal utilization of the allocated resources impact the performance of coverage algorithms. The details has been elaborated in Sect. 5.2.

The primary objective of this chapter is to provide the reader with the motivation of designing resource constrained coverage algorithms for real networks. At first, we present a systematic understanding of the underlying challenges involved in dealing with the inherent tradeoff between the coverage speed and the wastage of bandwidth related to such problems. Then we go on formulating the general problem definition and subsequently discuss in details the optimal algorithm (with zero walker memory) to solve it, covering our works in [27–29], which forms the major content of this chapter. Accordingly, the design of solution for optimal coverage for regular $d$-dimensional grids for $d > 2$ has been elaborated in details. However, nodes in real world networks do not always exhibit homogeneous degrees. To bridge this gap and to have completeness, in the later portion, briefly we discuss our work in [32] which extends the optimal algorithm and designs a solution to the problem assuming finite walker memory, considering more realistic resource constraints and present some results on random geometric graphs.

The problem of estimating random-walk-based coverage on graphs has been studied since 1951 by Dvoretzky and Erdos [11]. However, in [27] for the first time, we formulated the extended coverage problem which explores the challenge of designing optimal strategies with explicit control of resources (bandwidth and time consumption). Flooding and 1-RW can be trivially noted to be optimal under abundance of bandwidth and time, respectively. However the strategies in the

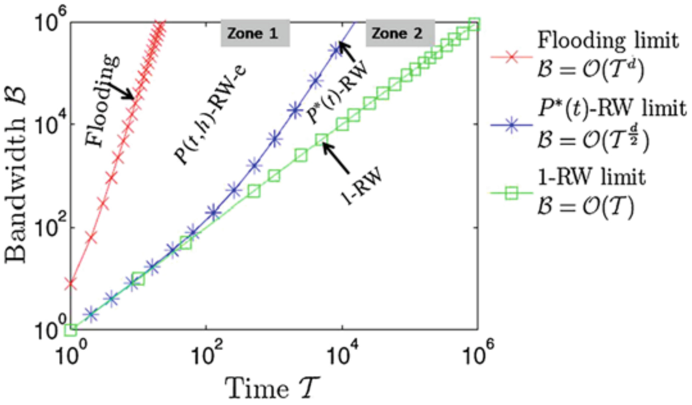


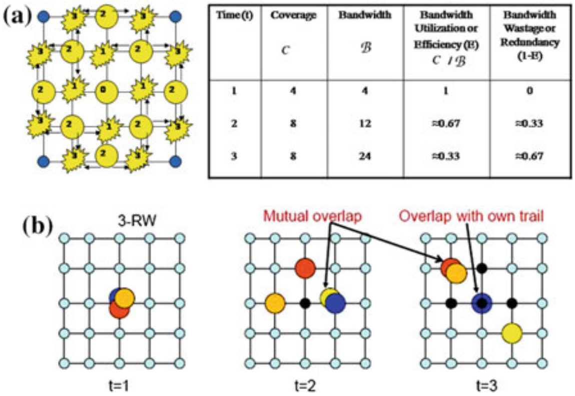
**Fig. 5.1** Phase diagram showing the zones in the bandwidth ($\mathcal{B}$) and time ($\mathcal{T}$) space, served by the strategies producing optimal coverage for the allocated resources of $\mathcal{B}$ and $\mathcal{T}$ towards execution of a service, on a d-dimensional regular grid network. The phase boundaries are annotated with the plot of asymptotic functions of bandwidth consumption with time, corresponding to a 3-dimensional grid, as an example. Flooding and 1-RW are optimal in the limit of large (unbounded) $\mathcal{B}$ and $\mathcal{T}$, respectively. The memory-less, uniform proliferation strategy $P^*(t)$-RW is optimal for time constraints $\mathcal{O}(\mathcal{B}^{\frac{2}{d}}) \leq \mathcal{T} \leq \mathcal{B}$, i.e., in Zone 2. The history-based($h$), non-uniform proliferation strategy $P(t, h)$-RW-e is near optimal in the time constraint range $\mathcal{O}(\mathcal{B}^{\frac{1}{d}}) < \mathcal{T} < \mathcal{O}(\mathcal{B}^{\frac{2}{d}})$, i.e., in Zone 2

intermediate range was unknown. There is a inherent tradeoff between coverage speed and bandwidth utilization (details in Sect. 5.2). However, for d-dimensional regular grid, we have shown that without any additional memory, in the problem space, the coverage algorithm $P^*(t)$-RW produces optimal coverage same as 1-RW (maximum achievable for an allocated bandwidth) at a much higher speedup. Our strategy is based on random walkers proliferating in a uniform rate. Also, a class of proliferating random walks $P(t)$-RW was shown be the best performing strategy in a generalized case. Later extending $P(t)$-RW, using small walker memory, in [32] we derived an near optimal strategy $P(t, h)$-RW-e for the generalized case. The findings are summarized in Fig. 5.1 where the application ranges of the strategies are compared in the phase diagram ($\mathcal{T}, \mathcal{B}$), by denoting the strategy which yields the maximum node coverage $C(\mathcal{T})$ at given ($\mathcal{T}, \mathcal{B}$).

The rest of the chapter is organized in six sections. Section 5.2 provides the background and problem definition. Sections 5.3, 5.4 and 5.5 discusses design of the strategies $P^*(t)-$RW, $P(t)$-RW and $P(t, h)$-RW-e. The related literature has been highlighted in Sect. 5.6. Finally, Sect. 5.7 concludes.

## 5.2 Background and Problem Definition

Due to the lack of an index and in absence of additional memory in the nodes (sites) or the messages, the coverage strategies either use some variants of flooding or random walk (RW). Let's consider two simpler cases of computing coverage: $[C(*, \mathcal{T})]$



| Time(t) | Coverage $C$ | Bandwidth $\mathcal{B}$ | Bandwidth Utilization or Efficiency (E) $C/\mathcal{B}$ | Bandwidth Wastage or Redundancy (1-E) |
|---|---|---|---|---|
| 1 | 4 | 4 | 1 | 0 |
| 2 | 8 | 12 | ≈0.67 | ≈0.33 |
| 3 | 8 | 24 | ≈0.33 | ≈0.67 |

**Fig. 5.2** Figure **a** illustrates the basic flooding process at time t = 3, starting from node labeled '0'. *Arrows* denote the direction of message forward. The *yellow circles* denote a node visit in some previous time (denoted by label) step. *Yellow* 'bursts' denote a redundant visit at the current step. The corresponding table shows the change in coverage, bandwidth, efficiency and redundancy at a particular time step. Here, redundancy increases fast with time. Figure **b** illustrates a 3-RW on a two dimensional grid. Three random walkers are shown in *bigger circles*. The *smaller circles* in *gray* and *black* represent respectively, unvisited and visited grid points. At time $t = 1$, all three walkers start from the source node. The mutual overlap (a walker revisiting some other walker's trail) and own overlap (revisiting its own trail) has been shown at $t = 2$ and 3

i.e. when there is no restriction in $\mathcal{B}$ and $[C(\mathcal{B}, *)]$ when there is no restriction in $\mathcal{T}$. When there is abundance of bandwidth $\mathcal{B}$, flooding produces optimal coverage in minimum time. Here, *optimal coverage* refers to the maximum possible coverage under given constraints of bandwidth and time. In *flooding*, a node forwards the copy of a received message to all its neighbors except one from which it received the message. Figure 5.2a illustrates flooding in a 2-D grid of 25 nodes. It shows that flooding maximizes speed of coverage (i.e. coverage per unit of time) at the cost of huge wastage of bandwidth. Flooding is a highly inefficient coverage process[1] as number of redundant visits increase exponentially with time. Hence, in many occasions flooding is not accepted in spite of its high speedup.

When there is no restriction of time, *single random walk (1-RW)* strategy produces optimal coverage with minimum wastage of bandwidth. In 1-RW, the message packet is modeled as a random walker, which originate from the start node, and spread around to serach/disseminate/gather the desired information. In each time step the walker is forwarded from its current node to a randomly selected neighbor node [2, 10]. The random walker produces a *overlap*, i.e.redundant visit of a node when it jumps onto its own trail. Hence, it runs for time $\mathcal{T} = \mathcal{B}$ units and is the most efficient strategy producing optimal coverage say, $C_1$, however with least speedup. Hence, for a given bandwidth $\mathcal{B}$ there is an inherent tradeoff between the coverage speed and the wastage of bandwidth. Let's now ask [27] a counter intuitive question:

[1] This chapter uses the term process, strategy and algorithm is used interchangeably.

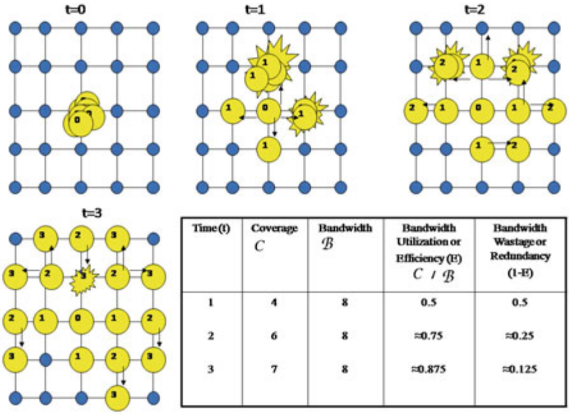
- **Fundamental Question**—*Given bandwidth quota $\mathcal{B}$, can we design a random walk strategy which produces the same coverage $(C_1)$ that of a 1-RW but with a higher speedup, i.e. consuming the allocated bandwidth in time $\mathcal{T} < \mathcal{B}$ units? In other words, can higher speedup be achieved without an increase in amount of bandwidth wastage?*

In an attempt to achieve both higher coverage speedup one may consider using $K(>1)$*-RW* [21], where $K$ number of independent random walkers start from the initial node to reach more distinct nodes than a single random walker can visit during the same time. The number of walkers remain fixed throughout the process. An illustration of 3-RW in Fig. 5.2b shows that overlaps in a multiple random walk-based strategy are of two kinds. First, *overlap with its own trail*, which is can't be avoided even if a single walker is in the system. Secondly, the *mutual overlap* in which two or more walkers visit the same node either in the same time step or in different time steps. Unlike multiple walkers, a single random walker only has its inherent overlap, hence, a K-RW will always be less efficient than 1-RW, therefore will fail to achieve the same coverage. Hence, given $\mathcal{B}$ there is an upper bound $C_{max}$ to the amount of achievable coverage which equals $C_1$, i.e., the coverage obtained by 1-RW. By using a larger $K$, the allocated bandwidth $\mathcal{B}$ can be consumed faster, however, at the cost of reduced coverage due to increased mutual overlap. Therefore, $K(>1)$-RW can't be the answer to the above question.

A further observation is that (refer Fig. 5.2) with time the $K$ random walkers disperse (move 'far' apart), reducing mutual overlap, however leaving unexplored area (unvisited nodes) in between. Hence, to simultaneously satisfy both objectives: first, to nullify/reduce the mutual overlap of a $K(\gg 1)$-RW and secondly, to ensure higher coverage speed, a *proliferating random walk* ($p(t)$-RW) [15] strategy may be considered. In a proliferating walk, a walker self-replicates at its current node with rate $p(t) > 0$ at time $t$, such that each walker produces another walker after every $1/p(t)$ time steps, on average. The walker count thus increases exponentially with time, hence, the bandwidth consumption too, increases exponentially. It may be noted that estimating the suitable non-zero proliferation rate is critical in designing a bandwidth constrained optimal coverage algorithm.

Previously developed algorithms have either set application-specific rules for walker proliferation or used a constant rate $p(t) = P_C$, where $P_C$ satisfies the condition $\mathcal{B} = \sum_{t=1}^{\mathcal{T}} K(t) = \sum_{t=1}^{\mathcal{T}} (1+P_C)^{t-1}$. Here $\mathcal{T}$ is the service time. To achieve our desired objective, starting with a small walker count $P_C$-RW and each proliferating at a constant moderate rate $P_C$, could be used. This, however, as demonstrated in this chapter (Sect. 5.3) by numerical simulation and quantitative comparison of different algorithms is a naive strategy, that neither guarantees elimination of mutual overlap (that might affect efficiency) in the early stage nor reduction of the uncovered area (that might affect speed) at the later stage (Fig. 5.3).

Noting the above fact, to solve the above-posed fundamental question, one now specifically has to solve the following problems:



| Time (t) | Coverage $C$ | Bandwidth $\mathcal{B}$ | Bandwidth Utilization or Efficiency (E) $C / \mathcal{B}$ | Bandwidth Wastage or Redundancy (1-E) |
|---|---|---|---|---|
| 1 | 4 | 8 | 0.5 | 0.5 |
| 2 | 6 | 8 | ≈0.75 | ≈0.25 |
| 3 | 7 | 8 | ≈0.875 | ≈0.125 |

**Fig. 5.3** Figure illustrates the $K = 8$-RW process for 3 time steps, all starting from node labeled '0'. *Arrows* denote the direction of message forward. The *yellow circles* denotes a visited node, label denotes the visit time. *Yellow* 'bursts' denote a redundant visit at the current step. The corresponding table shows the change in coverage, bandwidth, efficiency and redundancy at a particular time step. Here, redundancy decreases with time

- **Problem 1**: *Given $\mathcal{B}$, starting from a single node, without any memory of visited nodes, can $C_{max}$ be achieved much faster (compared to 1-RW) i.e., at $T = T_{min} \ll \mathcal{B}$ by multiple walkers using a suitably chosen proliferation strategy?*

In simple terms one need to design an optimal coverage strategy for $C(\mathcal{B}, *)$. The solution to the above network-challenge is not trivial even for regular grids. In Sect. 5.3 of this chapter, we study the problem for infinite $d$-dimensional Euclidean grids. With a deeper understanding of K-RW dynamics, in Sect. 5.3.2 we discuss how to design a uniform time varying random walk proliferation strategy ($P^*-$RW) [27] which ensures minimal wastage with an speedup of $S = \frac{B}{T_{min}} = \mathcal{O}({T_{min}}^{(\frac{d}{2}-1)})$. However, a real-world service operating with bandwidth $\mathcal{B}$ may need to complete even faster, i.e. within a time constraint $\mathcal{T} < T_{min}$. In such case, the random walk process is forced to consume the allocated bandwidth in less than optimal time. Hence process can deviate from the optimality criterion and afford to waste some bandwidth to achieve higher speedup, instead. As a result, $C(\mathcal{B}, \mathcal{T})$ will always be less than $C_{max}$. Hence, two important questions of practical relevance is:

- **Problem 2**: *Given $\mathcal{B}$ and $\mathcal{T}$, assuming zero memory of random walkers, what proliferation strategy optimizes coverage $C(\mathcal{B}, \mathcal{T})$, where $\mathcal{T} < T_{min}$?*

In Sect. 5.4, we show that for the above problem, in regular grids a random walk proliferation strategy $P(t)-$RW obtained by extending the optimal strategy $P^*-$RW performs the best compared to naive strategies $P_C-$RW and $\lceil \frac{\mathcal{B}}{\mathcal{T}} \rceil$-RW. However, the relative advantage of $P(t)-$RW decreases sharply as time constraint reduces, i.e. $\mathcal{T} \ll T_{min}$. Analysing the co-relation between mutual overlap and walker density, In [32] we developed a near-optimal strategy based on nonuniform random walk proliferation $P(t, h)-$RW-e which achieves 3.5 times more coverage than $P(t)-$RW. Here each walker is accommodated with a finite memory and $h$ is the number of mutual overlaps experienced by a walker in last $H$ hops. The brief description of $P(t, h)-$RW-e is presented in Sect. 5.5.

## 5.3 Coverage Strategy for $C(\mathcal{B}, *)$ Under Unconstrained Time

In their pioneering work Larralde et al. [21, 22] studied the $K$-RW dynamics on regular grids from a statistical mechanics perspective. It was the first attempt to visualize the dynamics of multiple $K \gg 1$ random walkers on an infinite $d$-dimensional Regular grid. They also derived an asymptotic expression for $C(\mathcal{T})$. Later it was supported by a more rigorous solution by Yuste et al. [41]. The design of our strategy takes insights from Larralde's work. Section 5.3.1 presents the $K$-RW dynamics on Regular grids and summarizes the results and important observations that we derive from their work. The optimal coverage strategy $P^*(t)$ is derived in Sect. 5.3.2 for d-dimensional regular grids. Section 5.3.3 presents the experimental verification of Larralde's results as well as optimality of $P^*(t)$. Finally the Sect. 5.3.4 draws the conclusion.

### *5.3.1 K-RW Dynamics on Regular Grids*

Figure 5.4 illustrates Larralde's experiment on the visualization of the surface geometry of spreading of $K$ independent random walkers all starting from a single node, on a 2-dimensional regular grid [22]. It shows that random walkers move far apart from each other as time increases. Table 5.1 summarizes the key analytical results derived in [21, 22, 41]. It shows that when $K$ random walkers all start from a single

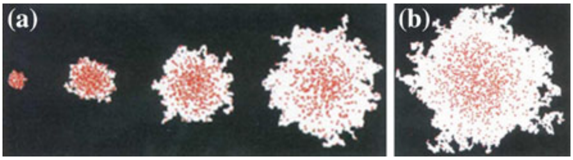


**Fig. 5.4** **a** Visualization of the actual set of sites (nodes) covered (visited) by $K = 500$ random walkers all starting from a single source in a two dimensional surface, at a sequence of four successive times $t$, showing the progressive roughening of the surface of this set as time increases. The set of visited sites are shown as *white*, individual random walkers shown in *red* and unvisited virgin territory is shown in *black*. **b** illustrates the case for $K = 1{,}000$ at a later time. The roughening of the surface is due to the fact that with increase in time random walkers go far apart from each other. (Figure courtesy [21, 22])

**Table 5.1** Table shows the expression for Coverage increments $\Delta C(t)$ achieved by $K \gg 1$ random walkers at a time step $t$ for three different time regimes in a $d(>2)$-dimensional regular grid [21, 22, 41] ($\xi^{'}$ and $\xi$ are parameters)

| **Coverage Regimes** : | **I** | **II** | **III** |
|---|---|---|---|
| | ==== | ============== | ==== |
| **Coverage** $C(t)$ : | $O(t^d)$ | $O(t^{\frac{d}{2}}[\ln(K\times t^{1-\frac{d}{2}})]^{\frac{d}{2}})$ | $O(K\times t)$ |
| | ==== | ============== | ==== |
| $\Delta C(t)$ : | $d\times t^{d-1}$ | $\frac{d}{2}[t\times\ln(K\times t^{1-\frac{d}{2}})]^{\frac{d}{2}-1}$ | $K$ |
| **Time** $(t)$: | $---$ | $-----------$ | $--->$ |
| | | $\Downarrow$ | $\Downarrow$ |
| **Crossover time**: | | $t_1^c(K)=\xi^{'}\times\ln K$ | $t_2^c(K)=\xi\times K^{\frac{2}{d-2}}$ |

It may be noted that for $d = 1, 2$, the $K$ random walkers show the same behavior in regimes I and II with identical $t_1^c$. But for $d = 1$ regime III does not occur and occurs at $t_2^c \sim e^K$ for $d = 2$

point on an infinite $d$-dimensional Regular grid the system of walkers exhibit three distinct regimes of spread. For each regime, the table gives the expressions for the asymptotic coverage at time $t$ ($C(t)$) and the *increase in coverage* at time $t$ ($\Delta C(t)$) during one time step and the crossover times from regime I–II ($t_1^c(K)$) and II–III ($t_2^c(K)$). It may be observed from the values, that as $t$ increases beyond the *crossover times* $t_1^c$ and $t_2^c$, the coverage increases uniformly but with qualitatively different behavior within each of the three regimes. Moreover the regime I is much short lived compared to regime II. An important to note that the analytical modeling of Larralde and Yuste is based on a key underlying assumption that walker count is large i.e., $K \gg 1$. Hence, for small vales of $K$ the exact expressions may not match although the qualitative nature is expected to hold true (details discussed in Sect. 5.3.3).

In regime I, all the walkers clutter together and the mutual overlap probability is very high. However, unvisited neighbors of all the visited nodes are reached during the next step, resulting in a flooding-like coverage. The considered $K$-RW enters regime II when $t$ passes $t_1^c$ as the walkers gradually move away from each other and less walkers co-occupy nodes. However, still some amount of mutual overlap persists. In regime III, the walkers are sufficiently separated such that mutual overlap almost vanishes. As a result, from time $t = t_2^c(K)$ onwards, i.e., *crossover* from regime II–III, each walker behaves independently like a single random walker with non-overlapping exploration space.

***Critical Observation***—We observe that $t = t_2^c(K)$ is an *optimal time* for efficient coverage since it yields a high coverage increment per time step due to low mutual overlap and low wastage of inter-walker space. The crossover time $t_2^c(K)$ is reached early if $K$ is small.

Let us denote by $E(t)$ the *efficiency* during time step $t$, which is defined as the ratio of the coverage increments, $\Delta C(t)$, to the number of walkers used during time step $t$. In regimes I and II efficiency, $E(t)$, increases with time but reaches and keeps the peak value $E(t \geq t_2^c) = E_{max}$ as it enters regime III. $E_{max}$ is the efficiency of a single random walker.

### 5.3.2 Strategy $P^*(t)$ for d-Dimensional Grids

With a constant number $(K)$ of walkers the system will cross over from regime II–III at a time

$$t_2^c(K) = \xi \times K^{\frac{2}{d-2}} \tag{5.1}$$

If no further walkers are introduced, the system will stay in regime III where overlaps do not decrease further. But at the same time coverage speed i.e., $\Delta C(t)$ remains constant with respect to time. Hence, large parts of the network remain unexplored. However, the initial overlap of $k$-RW during time $t < t_2^c(K)$ can be avoided by starting with a small number of random walkers at $t = 1$ and proliferating each walker at a suitable rate $P^*(t)$ at each time step. Thus the system can always remain at the regime II–III boundary as desired. The exact expression of $P^*(t)$ is derived next.

**Calculation of** $P^*(t)$—Let the per walker proliferation rate $P^*(t)$ produce $K(t)$ walkers at time $t$. Then $K(t)$ can be obtained from the following recurrence:

$$K(t+1) = K(t) \times (1 + P^*(t)) \tag{5.2}$$

We consider a slowly changing walker number $K(t)$ and extend the dependency of $t_2^c(K)$ on $K$ from Eq. 5.1 to $K(t)$ by adiabatic approximation and insert the requirement that $K(t)$ maintains the system at the regime boundary.

$$t_2^c(K) = t = \xi \times K(t)^{\frac{2}{d-2}} \Rightarrow K(t) = \left(\frac{t}{\xi}\right)^{\frac{d-2}{2}} \tag{5.3}$$

Hence, the initial walker count is $K(1) = \xi^{-\frac{d-2}{2}}$ which provides the *starting value* for the recurrence in Eq. 5.2. Substituting the values of $K(t+1)$ and $K(t)$ as obtained from Eq. 5.3 into Eq. 5.2, we get:

$$\left(\frac{t+1}{\xi}\right)^{\frac{d-2}{2}} = \left(\frac{t}{\xi}\right)^{\frac{d-2}{2}} \times (1 + P^*(t)) \tag{5.4}$$

Hence by simplifying the above Eq. 5.4, $P^*(t)$ is obtained as following:

$$P^*(t) = \left(1 + \frac{1}{t}\right)^{\frac{d-2}{2}} - 1 \tag{5.5}$$

Expanding $P^*(t)$ in powers of $\frac{1}{t}$ and ignoring the higher order terms in $\frac{1}{t}$ in the asymptotic range $t \gg 1$, the proliferation rate takes the form

$$P^*(t) \approx \frac{d-2}{2} \times \frac{1}{t} \tag{5.6}$$

showing a fast decay with time. Hence, $K(t+1) - K(t) \ll 1$ and the adiabatic approximation in Eq. 5.3 is consistent.

Since $P^*(t)$-RW consumes the bandwidth $\mathcal{B}$ in $T_{min}$ time steps, we can write $\mathcal{B} = \sum_{t=1}^{T_{min}} K(t)$, where substituting $K(t)$ from Eq. 5.3 we get:

$$\mathcal{B} = K(1) \times \sum_{t=1}^{T_{min}} t^{\frac{d-2}{2}} \tag{5.7}$$

We estimate $\mathcal{B}$ by approximating the discrete sum of bandwidth consumption in each time step as an integral and solving it, as following:

$$\begin{aligned} \mathcal{B} &\approx \int_{1}^{T^{min}} \left(\frac{t}{\xi}\right)^{\frac{d-2}{2}} dt \\ &= \left(\frac{1}{\xi}\right)^{\frac{d-2}{2}} \left[\frac{t^{\frac{d}{2}}}{\frac{d}{2}}\right]_{1}^{T^{min}} \\ &= \left(\frac{1}{\xi}\right)^{\frac{d-2}{2}} \times \frac{2}{d}\left[T^{min\frac{d}{2}} - 1\right] \end{aligned} \tag{5.8}$$

**Calculation of Speed-up** $S$—The strategy $P^*(t)$-RW utilizes the given bandwidth $\mathcal{B}$ in a best possible way, consuming it in least time $T_{min} = \mathcal{O}(B^{\frac{2}{d}})$, providing maximal possible coverage $C_{max}$. $T_{min}$ is the lower bound on the time to consume $\mathcal{B}$. Given $\mathcal{B}$, the ratio of the time taken to obtain $C_{max}$ by 1-RW to the time taken by $P^*(t)$-RW is defined as the service *speed-up*, $S$ expressed as:

$$S = \frac{B}{T_{min}} = \mathcal{O}(B^{\frac{d-2}{d}}) = \mathcal{O}(T_{min}{}^{(\frac{d}{2}-1)}) \tag{5.9}$$

It may be noted that for calculating the values of $S$ and $P^*(t)$, an empirical estimation of $\xi$ is needed. We present an empirical estimation of $\xi$ in next subsection. In addition Sect. 5.3.3 provides an empirical verification of Larralde's results and also the approximative result for the service speed-up and the performance of the proliferation rate $P^*(t)$-RW.

### 5.3.3 Empirical Verification

A regular torus grid has been used in our simulations, with large number of nodes so that the observed random walk is free from boundary (along a dimension) effects. Here the main challenge is to store the node adjacency information of a sufficiently large grid. A grid size of $\mathcal{L}$ has been chosen for simulation. Correspondingly for grid dimensions, $d$=3, 4, 5 and 6, $\mathcal{L}$ is chosen to be respectively $625^3$, $130^4$, $48^5$ and $25^6$. To store the grid structure, an array of size $\mathcal{L}$ is used, that equals the size of the grid. Each node has $2 \times d$ neighbor nodes. Hence storing the adjacency information of the entire topology requires huge memory. To cope up with the memory limitations, we have maintained the adjacency list data structure as following. Each element of the array is a record (typically consuming 24 bytes storage), corresponding to a grid node with one field storing the address of the neighborhood list (dynamically allocated), with respective tags storing the information whether the neighbor is visited or not. Before starting each simulation run, we initialize all the neighbor list entries to NULL and allocate their space only when a particular node gets visited. At the starting of the next run we de-allocate all the neighborhood list entries. Let's consider as an example $d = 4$, here $\mathcal{L} = 25 \times 10^8$, hence the initialized node array occupies 600 MB (approx) space. For chosen values of B=100,000, at the end of a run, the total space consumed by the allocated neighborhood lists is ($10^5 \times 8 \times 4 \times 2$) (approx.) = 6.4 MB, as each node has entry corresponding to 8 neighbors storing neighbor-id and visit tag for each and a long INT consumes 4 bytes. Hence the total space consumption in each run is restricted to 610 MB (approx.). We used a machine with 4 GB RAM. All simulation data were averaged over 10,000 realizations.

The simulation results are directed towards identifying the following facts:

1. Existence of three distinct regimes for any $K$-RW process where $K > 1$.
2. Identification of the crossover time from regime II–III $t = t_2^c(K)$ which is important as it is an optimal time for obtaining efficient coverage and also used to estimate $\xi$.
3. Verifying the optimality of $P^*(t)$.

**Observing the regimes**—To identify the regimes we plot the efficiency $E(t)$ across time $t$ for a $K > 1$-RW process. Figure 5.5a shows the lin-log scale plot of the efficiency $E(t)$ during the time step $t$ versus $t$ for $K = 1$ and 3 for $d$ =3, as a case. The presence of three distinct regimes has been shown. Regime I is observable in the 1st step. Regime II is observed by decreasing difference in efficiency with compared to single random walker. Regime III is denoted by the time during which the efficiency almost matches with that of a single walker with a steady small non-zero difference. The non zero difference is due to the stochastic nature of the experiment. Figure 5.5b illustrates the crossover time from regime II–III for $K = 3$ ($t_3^c = 1,620$) which is estimated from the intersection of the asymptotic of regime II (decreasing difference) and regime III (fluctuating difference around a constant nonzero average). Similar behavior has been observed for any $K > 1$ across different dimensions $d$ =4, 5

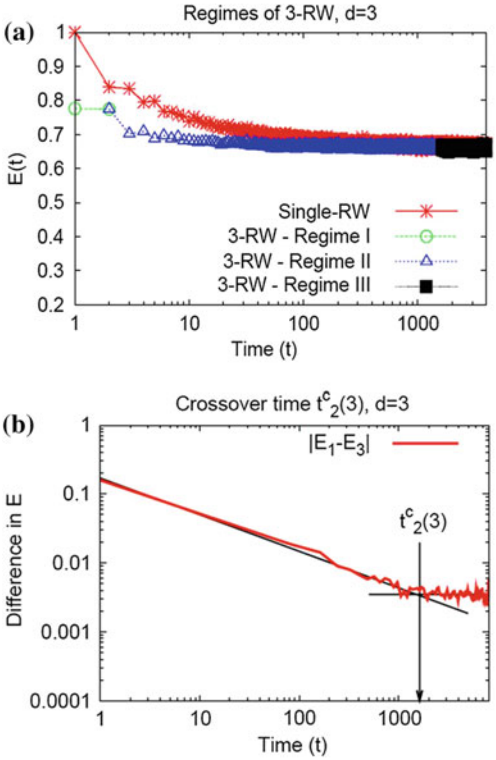


**Fig. 5.5** **a** Efficiency $E(t)$ versus time $t$ (on lin-log scale) for $K$-RW on a 3-dimensional Euclidean grid of $|L| = 1{,}304$ nodes. $K = 1$ is shown in *red star* throughout. Three regimes for $K = 3$ are shown, regime I in *green circles*, regime II in *blue triangles* and regime III in *black boxes* which occur overlayed on *red stars*. For 3-RW the crossover time from regime II–III ($t_2^c(3)$= 1,620) is estimated from the intersection of the asymptotic (denoted by *solid black lines*) of regime II (decreasing difference) and regime III (fluctuating difference around a constant nonzero average), as shown in figure (**b**) by a log-log plot of $|E_1 - E_3|$

and 6. The above plots provides a qualitative confirmation of the results in Table 5.1 even when $K$ is relatively small.

**Empirical estimation of** $\xi$—For a given $K$, we go on estimating $t_2^c(K)$ empirically from the plot of difference in efficiency $|E_1 - E_K|$ versus $t$. $K$-RW simulations for different discrete values of $K$ are performed. For grid $d = 3$ as a case, the values of $t_2^c(K)$ are recorded for different walker number, $K = 1$–12 at an interval of 1, and then a least square fit of Eq. 5.1 to the values is made. The fit yields $\xi = 176.12$. Figure 5.6 presents the plot of recorded $t_2^c(K)$ values and the fitted curve. For grid dimensions $d = 4$, 5 and 6, experiments produced the values of $\xi = 96$, 30 and 18 respectively, as used in our work.

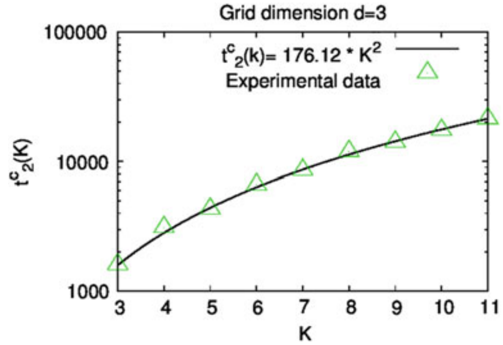


**Fig. 5.6** Figure shows the plot of recorded $t_2^c(K)$ values versus random walker count $K$ as '$\Delta$' symbols for grid $d = 3$, as an example. The *solid line* illustrates the curve obtained as a result of least square fit of equation $t_2^c = \xi K^2$ (refer Eq. 5.1) to the data, which yields the value of $\xi = 176.12$

Theoretically, as $\xi > 1$ for grid dimensions $d = 3 \ldots 6$, it implies that the initial walker number $K(1) < 1$. However, in order to run the simulation, the coverage process has to start with at least one random walker. To nullify this discretization effect we modify $P^*(t)$ to $P^*(t + t')$ where $t'$ is the earliest time when $K(t')$ just becomes $\geq 1$, if simulated for continuous $K$ with $P^*(t)$ and $K(1) = \xi^{-\frac{d-2}{2}}$. Substituting $K(t = t') = 1$ in Eq. 5.3 yields $t' = \xi$. $P^*(t + t')$ ensures that $\Delta K(t)$, the increase in walker count, remains the same as desired by the analytic rate in Eq. 5.5, even if the coverage starts with $K(1) = 1$.

**Verification of the optimality of $P^*(t)$ (in terms of efficiency and time)**—To have a through understanding of the behavior of $P^*(t)$ to know how the proliferation rate impact the functioning within the different regimes, we extend Eq. 5.5 as following:

$$P(t) = \left(1 + \frac{1}{t + \xi - 1}\right)^{(\alpha \times \frac{d-2}{2})} - 1 \tag{5.10}$$

Two modifications are made. First, we included a new parameter α in the exponent term. The parameter α tunes the degree of freedom of the coverage process for exploring new nodes within regimes II–III. This suitable choice of the value of $\alpha$ lets us control in which regime the process should operate. When $\alpha > 1$, process functions within regime II with constant efficiency $E < E_{max}$. When we choose $\alpha < 1$, it lets the system function within regime III with constant efficiency $E_{max}$. In both cases the system operating point correspondingly maintains a constant distance from the regime II–III boundary. Secondly, the $\xi$ term is included to consider at least one initial walker at the start of the coverage process. We denote $P(t)|_{\alpha=1,\xi=1}$ as $P^*(t)$.

The optimality of $P^*(t)$ is derived analytically through Eqs. 5.3–5.5. The derivations tell that, theoretically, using $\alpha = 1$, one can attain maximum possible coverage in minimum time. Hence, it is expected that the coverage of $P(t)$-RW is exactly similar to that of 1-RW for $\alpha \leq 1$, whereas coverage should sharply fall as $\alpha$ is increased beyond 1. Figure 5.7 shows the relative difference in coverage between

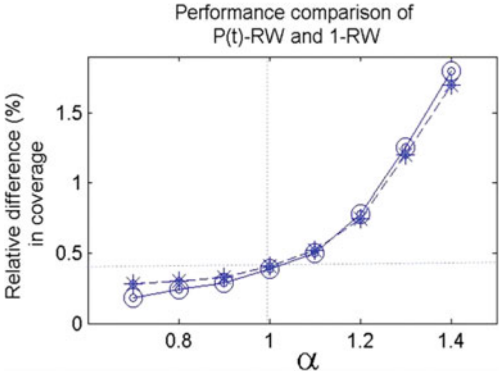


**Fig. 5.7** The plot shows the relative difference in coverage $\frac{C_{max}-C(T)}{C_{max}} \times 100\,\%$ obtained using the generalized coverage algorithm $P(t)$-RW with respect to 1-RW for $d = 3$ and 6 dimensional regular grid for different values of α. The simulations considered allocated bandwidth of $\mathcal{B} = 5 \times 10^4$

$P(t)$-RW and 1-RW and reveals that it is as low as 0.3 % for $\alpha < 1$, whereas it increases significantly for $\alpha > 1$, for grid dimensions $d = 3$ and 6. Qualitatively similar results hold for dimensions 4 and 5. Hence in experimental validation we could not exactly reproduce the theoretical result. There is a very small difference albeit a difference, for $\alpha \leq 1$, which is due to the stochastic nature of the experiment. Further the absence of abrupt rise in difference in coverage for $\alpha > 1$ stems from the smooth transition between regimes II ($\alpha > 1$) and III ($\alpha < 1$).

That is why, we look further into the dynamics of the random walkers around that region $\alpha = 1$ and show that it becomes much more difficult to cover new area beyond the optimal strategy. We study the behavior of $P(t)$-RW (Eq. 5.10) as a function of α taking $d = 3$ as a case study. For a given constraint in $\mathcal{B}$, the performance of $P^*(t)$ can be assessed in the context of the coupled optimization problems of latency $T$ and coverage $C$. The performance is compared with respect to to 1-RW. Figure 5.8 presents results on dimensions 3 and 6. From Fig. 5.8a we observe that efficiency decreases fast if one tries to consume the given $\mathcal{B}$ slightly faster by increasing α beyond 1. On the other hand, in order to increase the efficiency slightly further than that achieved by $P^*(t)$, by decreasing α below 1, one would have to spend a much longer time $T$ (Fig. 5.8b). The same behavior has been found for other investigated grid dimensions $d = 4$ and 5. Hence the solution space is multi-dimensional which requires design of an objective function that combines both factors, time and the efficiency.

**Verifying optimality of $P^*(t)$ (in terms of combined metric $M(\alpha)$)**—For quantitative assessment of the coupled optimization problem in a multi-dimensional solution space, a metric is required that combines time and efficiency. Among several possible combinations, we consider the combined product metric $M$ as:

$$M(\alpha) = T(\alpha) \times (E_{max} - E^{avg}(\alpha)) \tag{5.11}$$

$M$ combines the time needed to consume the allocated bandwidth (first factor) with a measure for the wastage of bandwidth ($E_{max} - E^{avg}(\alpha)$) as compared to the

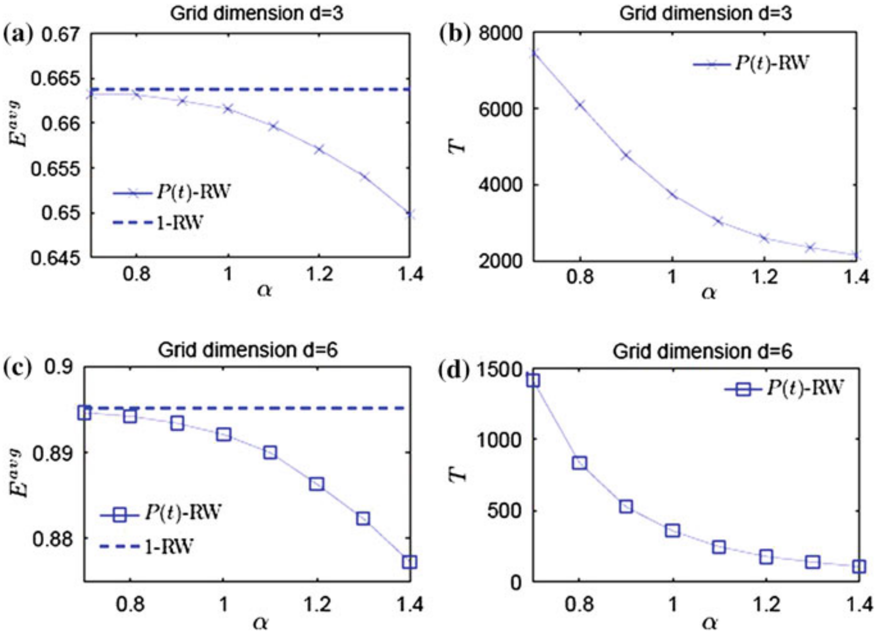


**Fig. 5.8** Performance of the generalized coverage algorithm $P(t)$ for different α, for $d = 3$ and 6 considering allocated bandwidth $\mathcal{B} = 1 \times 10^4$, as a typical case. **a** and **b** show plot of mean efficiency $E^{avg} = \sum_{t=1}^{T} \frac{E(t)}{T}$ and the time $T$ taken by a $P(t)$-RW to consume the given bandwidth $\mathcal{B}$, respectively, for different values of α. The *dashed lines* corresponds to $E_{max}$ and equals the mean efficiency of a 1-RW

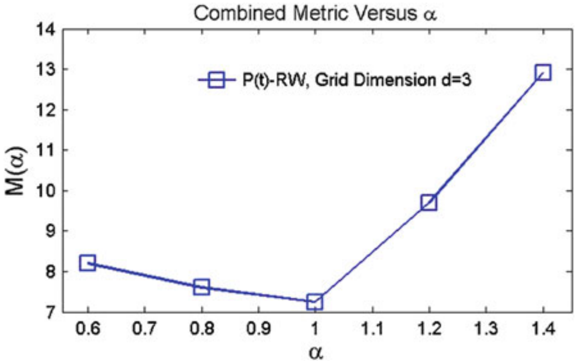


**Fig. 5.9** Plot of Combined metric $M(\alpha) = T(\alpha) * (E_{max} - E^{avg})$ as function of α for $d = 3$ as a case

maximally achievable efficiency, $E_{max}$ of the 1-RW. The desired algorithm shall operate fast and waste little, hence it *minimizes* $M$. Figure 5.9 shows the plot of $M(\alpha)$ for grid dimension $d$ =3. $M(\alpha)$ gets moderate values for small α (regime III), where a few walkers take a lot of time. On the other hand $M(\alpha)$ gets relatively large

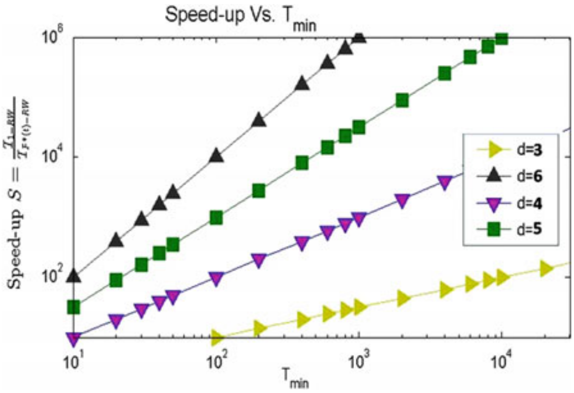


**Fig. 5.10** Plot of analytically derived Speed-up $S = \frac{B}{T_{min}} = \mathcal{O}(T_{min}{}^{(\frac{d}{2}-1)})$ versus the minimal time $T_{min} = T_{P^*(t)-RW}$ to consume the given bandwidth $\mathcal{B}$ for different dimensions $d = 3, 4, 5$ and 6 of an regular grid

values for large α (regime II), where many walkers waste bandwidth due to mutual overlap. It may be significant to note that in particular, $M(\alpha)$ increases rapidly for $\alpha > 1$ as the efficiency decreases rapidly beyond $\alpha = 1$. The *minimum* of $M(\alpha)$ is found at $\alpha = 1$. These numerical results confirm, that within the class of generalized algorithms given by Eq. 5.10, P*(t) with $\alpha = 1$ provides the best performance as measured by the product metric of 5.11. The same behavior has been found for all investigated network dimensions, $d = 4$, 5 and 6.

**Verifying speed-up**—Figure 5.10 plots the analytical result for the service speed-up (Eq. 5.9). The speed-up is significant and it has been found (refer [27] for details) that the theoretical and simulation values of $S$ match well except for small $T_{min}$ because, there the approximation of Eq. 5.5 by Eq. 5.6 fails.

### *5.3.4 Remarks*

As it is found that in absence of additional memory, for a given $\mathcal{B}$ when there is no restriction in service time proliferating random $P^*(t)$-RW provides optimal node coverage (=$C_1$) in time $T_{min}$; which solves the 1st problem of Sect. 5.2. Here, to achieve the same coverage of 1-RW but with a significant speedup, starting with a single walker, in each time step all random walkers present in the system need to proliferate at a time dependent uniform rate $P^*(t)$. However, in many real-world network applications the demand for time of completion of service $\mathcal{T}$ may be much smaller than $T_{min} = \mathcal{O}(\mathcal{B}^{\frac{2}{d}})$. Considering this fact in the next section we discuss about a strategy that achieves optimal coverage $C(\mathcal{B}, \mathcal{T})$ when $\mathcal{T} \ll T_{min}$; which solves the 2nd problem.

## 5.4 Strategy for $C(\mathcal{B}, \mathcal{T})$ with Zero Walker Memory

This section seeks solution to the second problem raised in Sect. 5.2, to find the best strategy to maximize coverage, when there is both bandwidth and time $\mathcal{T} < T_{min}$ constraint, under the assumption that no overlap history is maintained by the random walkers. The probable strategies and their performance evaluation is presented in Sects. 5.4.1 and 5.4.2, respectively.

### 5.4.1 *Coverage Strategies $P(t)$–RW($\alpha > 1$), $\lceil\frac{\mathcal{B}}{\mathcal{T}}\rceil$-RW and $P_C$-RW*

$P(t)$-RW designed in Sect. 5.3.3 actually refers to a class of coverage strategies where $P(t)$-RW at $\alpha = 1$ is actually $P^*(t)$-RW, the optimal strategy to consume the bandwidth quota $\mathcal{B}$. Hence, in a scenario, when time constraint $\mathcal{T} < T_{min}$ is given, $P^*(t)$-RW, if used, will result in some excess (unutilized) bandwidth of amount $B_e = \sum_{t=\mathcal{T}}^{T_{min}} (\frac{t}{\xi})^{\frac{d-2}{2}}$ left even after allocated time limit has elapsed. It has been observed in Sect. 5.3.3 that $\alpha$ controls the speed of bandwidth consumption in $P(t)$ (refer Eq. 5.10). $\alpha > 1$ consumes $\mathcal{B}$ faster than $T_{min}$. Hence, *to ensure complete utilization of bandwidth within time limit, as a natural extension one may use proliferating random walk $P(t)$-RW, with a suitably chosen value of* $\alpha > 1$ *in such a scenario.*

Apart from $P(t)$-RW($\alpha > 1$), two more naive random walks may be used. First, $\lceil\frac{\mathcal{B}}{\mathcal{T}}\rceil$-RW, which use multiple $K = \lceil\frac{\mathcal{B}}{\mathcal{T}}\rceil$ random walkers all starting from the start point. Secondly, $P_C$-RW, which use random walkers with a constant proliferating rate independent of time. Here, $P_C$ is estimated empirically such that $\mathcal{B} = \sum_{t=1}^{\mathcal{T}} K(t-1) \times (1 + P_C)$, and $\mathcal{B}$ just gets consumed within allocated time $\mathcal{T}$. We assume $K(1) = 1$. A comparative analysis of all the three strategies is given next.

### 5.4.2 *Performance Evaluation*

The performances of $\lceil\frac{\mathcal{B}}{\mathcal{T}}\rceil$-RW, $P_C$-RW and $P(t)$-RW are compared in Fig. 5.11 with K=$\lceil\frac{\mathcal{B}}{\mathcal{T}}\rceil$, $P_C$ and $\alpha$ chosen such that $\mathcal{B}$ was consumed within $\mathcal{T}$. Simulations are done with $\mathcal{B}$ =50,000 and 25,000 for $d$=3 as a case. Figure 5.11a, b show the performance of $P(t)$-RW to be superior, especially when $\mathcal{T} \ll T_{min}$, where $P(t)$-RW yields almost 10 and 20 % more coverage than $\lceil\frac{\mathcal{B}}{\mathcal{T}}\rceil$-RW and $P_C$-RW, respectively. Simulation results for other dimensions $d = 4$, 5 and 6 also exhibit similar behavior. The performance of $P(t)$-RW improves for higher bandwidth and lower dimensionality. The results imply that the class of proliferating strategies $P(t)$-RW can suitably control the proliferation rate to produce more coverage when, compared to the time, there is an abundance of bandwidth, which however is not as much as required for flooding. However, from Fig. 5.11b it may be noted that as $\mathcal{T}$ decreases the

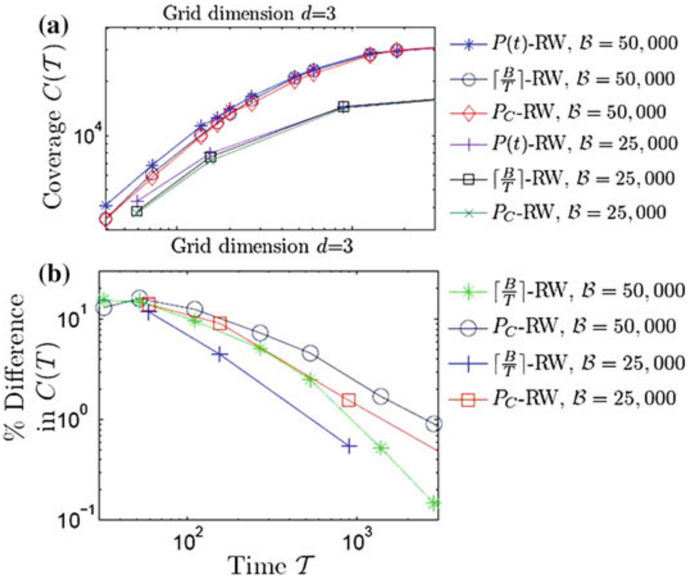


**Fig. 5.11** **a** Shows coverage $C(\mathcal{T})$ versus $\mathcal{T}$ for $P(t)$-RW, $\lceil\frac{\mathcal{B}}{\mathcal{T}}\rceil$-RW and $P_C$-RW, with $d = 3$ as a case, for $\mathcal{B} = 5 \times 10^4$ and $25 \times 10^3$. **b** Shows relative difference in $C(\mathcal{T})$ versus $\mathcal{T}$ obtained respectively, by $P_C$-RW and $\lceil\frac{\mathcal{B}}{\mathcal{T}}\rceil$-RW compared to $P(t)$-RW. For $\mathcal{T} \ll T_{min}$, $P(t)$-RW yields almost 10 and 20 % more coverage than $\lceil\frac{\mathcal{B}}{\mathcal{T}}\rceil$-RW and $P_C$-RW, respectively. However, for high values of $\mathcal{T}$, the difference among the strategies are insignificant

advantage of $P(t)$-RW over $\lceil\frac{\mathcal{B}}{\mathcal{T}}\rceil$-RW and $P_C$-RW reduces. A study of the dynamics of $P(t)$-RW is done next to investigate the reason.

**Analyzing the dynamics of $P(t)$-RW for** $\alpha > 1$—The system dynamics when $P(t)$-RW is used with values of $\alpha > 1$ is presented in Fig. 5.12. Plots in Fig. 5.12a show that when $\mathcal{T} < T_{min}$ and $\alpha > 1$, after the initial transient the efficiency of the $P(t)$-RW remains steady, however at a lower value compared to the efficiency of $P^*(t)$. The plot in Fig. 5.12b shows a nearly constant difference in efficiency $|E_{\alpha=1} - E_{\alpha=1.5}|$ in asymptotic time range. It implies that the coverage process operates at a point in regime II whose distance from regime II–III boundary remains fixed. However, when $\mathcal{T} \ll T_{min}$, then $\alpha \gg 1$. For relatively large values of $\alpha$ compared to 1, the efficiency falls sharply with time. As a result the difference in efficiency plot exhibits a positive slope (Fig. 5.12d). It implies that when $\alpha \gg 1$, the strategy $P(t)$ fails to operate at some point in regime II whose distance remains fixed relative to the regime II–III boundary, rather with time the operating point gradually shifts deep into regime I. Hence, experiments shows that as long as $\alpha$ is not too great, the system stabilizes at some point in Regime II (but relatively close to the boundary of Regime II–III) and maintains a constant efficiency, implication being that $P(t)$-RW($\alpha$ 1) tries to maintain an near optimal position.

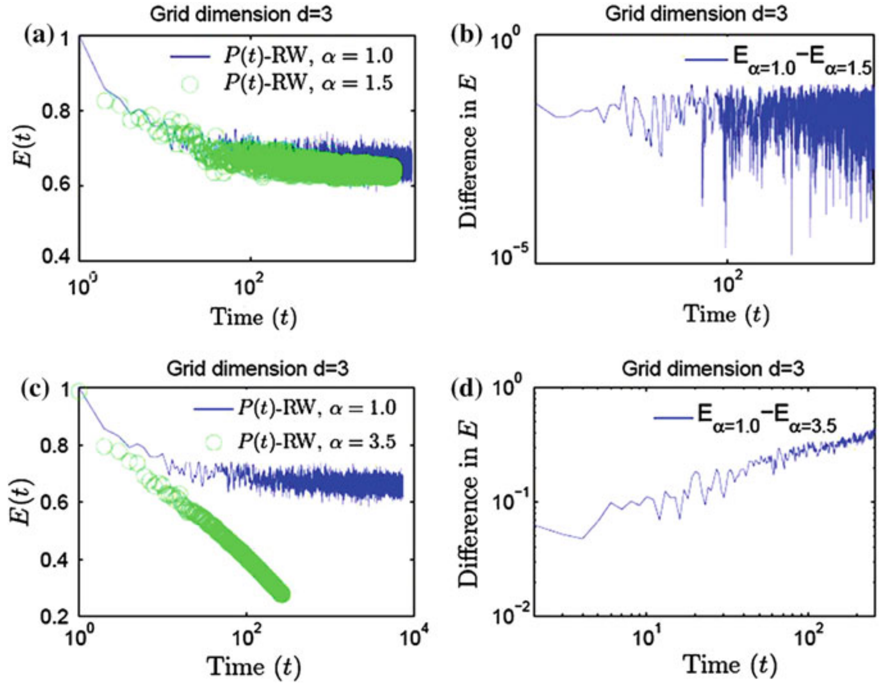


**Fig. 5.12** Figures show, given $\mathcal{B}$ =50,000 in $d = 3$ as a case, using $P(t)$-RW, how the allocated time $\mathcal{T}$ affects the coverage efficiency. **a** and **c** plots efficiency $E(t)$ versus time (t) in lin-log scale for $\alpha = 1.5$ and $\alpha = 3.5$ respectively, along with $\alpha = 1.0$. **b** and **d** show the difference in efficiency plot versus time in log-log scale for $\alpha = 1.5$ and $\alpha = 3.5$ respectively. As recorded the time taken to consume $\mathcal{B}$ is $\mathcal{T} = T_{min}$ = 7413, 4600 and 250 respectively for $\alpha = 1.0$, 1.5 and 3.5. As observed $E(t)$ remains steady in (**a**) when $\mathcal{T} << T_{min}$ however it falls drastically with time in (**c**) when $\mathcal{T} < T_{min}$

### *5.4.3 Remarks*

Although $P(t)$-RW for $\alpha > 1$ is found to be the best performing strategy for maximizing $C(\mathcal{B}, \mathcal{T})$ in absence of walker and node memory, the optimality of the strategy has not been proved. Moreover, the efficiency of P(t)-RW falls very rapidly as the constraint on time become stricter. As $\mathcal{T} \ll T_{min}$ the system functions at some point far from the regime II–III boundary, hence random walkers experiences higher mutual overlap as well higher walker density. However, it may be intuitively understood that density is directly related to their mutual overlap and that all walkers may not experience similar amount of mutual overlap. It triggers the motivation to use a different approach altogether, of adding small memory to random walkers so that they can proliferate in a nonuniform rate based on the amount of mutual overlap they experience, which has been explored in [32]. In the next section, briefly we present the work.

## 5.5 Strategy for $C(\mathcal{B}, \mathcal{T})$ with Finite Walker Memory

The main drawback of $P(t)-$RW is that, although the best performer with respect to naive strategies, it was not shown to be optimal. Secondly, it is highly inefficient and performs miserably under stricter time constraints. In this section we present our work in [32] which overcomes these limitations. [32] investigates the problem: *Assuming walkers have small memory, starting from a single node, design a random walk that maximize the number of distinctly visited nodes, i.e., coverage* $(C(\mathcal{B}, \mathcal{T}))$, *when the system operates with similar dynamics as that of regime II of the K-RW process*. In contrast to the approaches discussed till now, they proposed that each walker maintains the *mutual overlap history* they experience. The work is inspired by the deeper understanding of the dynamics of mutual overlap in regime II, discussed in Sect. 5.5.1. Section 5.5.2 presents the design of a nonuniform proliferating random walk strategy $P(t, h)$-RW where $h$ measures the number of mutual overlap. The performance evaluation of the strategy has been shown in Sect. 5.5.3.

### *5.5.1 Dynamics of Mutual Overlap*

The objective is to analyze the amount of mutual overlap experienced by each random walker present in the system at a particular point of time. To record the count of mutual overlaps, each random walker is equipped with a finite memory of size $H$. Let during the last $H$ visits a walker encounters $h$ mutual overlaps. It is intuitive to understand that walker density is related to their mutual overlap. The dynamics has been studied under the above postulate. Hence, the value of $h$ will provide a measure (temporal approximation) of the spatial density of other walkers present in the walker's current proximity. To estimate the walker density at different places throughout the system, the whole system, a regular grid of dimension 3 is divided in an arbitrary fashion into small cubes of size $10 \times 10 \times 10$. The walker count in each cube in then counted. To understand the dynamics, the system is restricted to operate with dynamics similar to regime II of $K$-RW, hence, $P(t)-$RW with $\alpha \gg 1$ is considered. Figure 5.13a plots the walker density for $P(t)-$RW, varying $\alpha$ and the developed strategy $P(t, h)-$RW (discussed in Sect. 5.5.2). The standard deviation of the mutual overlap among the walkers using is relatively high. These results reveal that *a large fraction of the area containing walkers stays sparse* in walker density and *there exists a huge heterogeneity in the density of the walkers when a walker proliferates at a higher rate* to gain more speedup.

Figure 5.13b shows the plot of mutual overlap versus walker density level for the $K = 50$-RW in the regime II. It reveals that exists two considerably different mutual-overlap characteristics with density states denoted as: d-low (low density) and d-high (high density). It may be noted that the mutual-overlap level in d-low-state is very low. However, as density increases there is a sudden changes to a d-high state, where the mutual overlap is high and remains almost constant, irrespective of minor changes

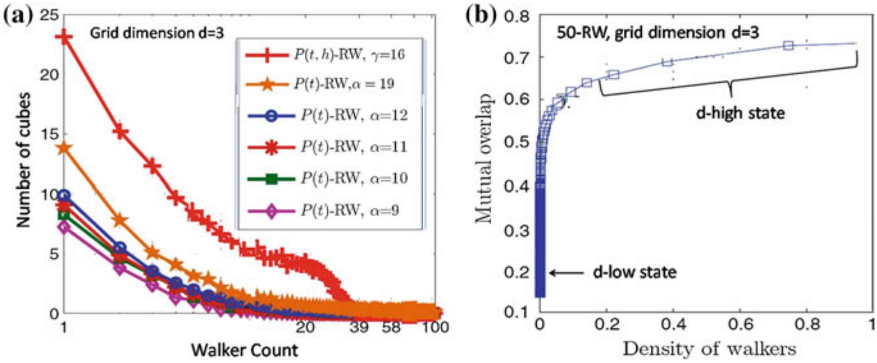


**Fig. 5.13** **a** Lin-log plot of the number of cubes (10 × 10 × 10 node sized) containing walkers in a 3-dimensional regular grid ,considering $P(t)$-RW for $\alpha = 9$, 10, 11, 12 and the designed strategy P(t,h)-RW (refer Sect. 5.5.2) with $\gamma = 16$ and $H = 20$. Comparisons are made at the 200th time step under constraints $\mathcal{T} = 200$ and $\mathcal{B} = 10^5$ units. **b** The plot of mutual overlap versus density level using $K = 50$-RW for 1,000 time steps obtained from simulations. (Figure courtesy [32])

in the density level. It may be trivially noted that it is the walkers which operate in the d-low state that contribute significantly to overall coverage achieved.the walkers at the d-high state has negligible impact in coverage speed-up.

### 5.5.2 Design of History-Based Proliferation $P(t, h)$-RW

The history-based proliferation strategy is designed based on the following facts observed in the previous subsection: (a) the amount of mutual overlap is a nonlinear increasing function of walker density, (b) over time, due to the probabilistic nature of spread of walkers, the walker density in the system does not stay uniform over all places and (c) walkers only in the d-low state contribute significantly to the increase in coverage at each instant. Hence, an optimal strategy to proliferate need to maximize the number of walkers in the d-low-state. A simple way through which walkers can identify themselves as d-low-state walkers is by tracking the ratio of $\frac{h}{H}$.

The required strategy can be summarized as follows [32]: *For maximal coverage, choose a decent history size H for each walker and proliferate only those walkers which have not encountered any walker in the last H node visits.*

The history-based proliferation strategy $P(t, h)$-RW has been designed by extending Eq. 5.10. The non-uniformity in the proliferation is achieved by replacing α as following:

$$\alpha_h = \alpha \times (1 - \frac{h}{H})^{\gamma} \tag{5.12}$$

Here, the parameter $\gamma$ controls the variation of the proliferation rate for different levels of mutual overlap faced by the walkers. For $\gamma = 0$, the strategy is same as $P(t)$-RW. Higher values of $\gamma$ would trigger proliferation of walkers having low mutual overlap ($h$ value).

### 5.5.3 Performance Evaluation of $P(t, h)$-RW and $P(t, h)$-RW-e

The plots in Fig. 5.14 measure coverage values under different time constraints and $\mathcal{B} = 10^5$. A huge improvement ($\approx$2 times more coverage) is noted by $P(t, h)$-RW ($\gamma = 16$, $H = 20$) over $K$-RW, $P_C$-RW and $P(t)$-RW. $P(t, h)$-RW was found to perform much better than $P(t)$-RW. A closer investigation shows that the superior performance of $P(t, h)$-RW is attributed to the fact that, here, relatively higher number of random walkers operate in d-low-state compared to other strategies.

**Optimality of parameters $\gamma$ and $H$**—It has been observed that at $\gamma = 0$ both $P(t)$-RW and $P(t, h)$-RW behaves the same as expected, however the performance benefits of $P(t, h)$-RW is more prominent as $\gamma$ increases. It has been found that highest coverage has been achieved for $\gamma = 16$. It is also found that for $\gamma \geq 16$, actually, those walkers which get zero mutual overlap (i.e., $h = 0$) in the last $H$ visits are proliferated. The version of $P(t, h)$-RW with $\gamma \geq 16$ is denoted as $P(t, h)$-RW-e.

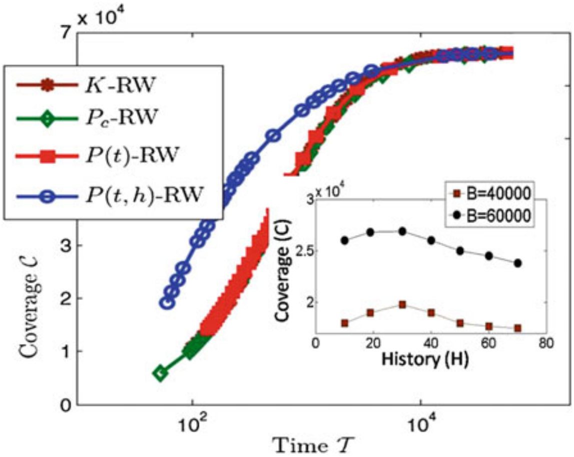


**Fig. 5.14** Plot of the coverage achieved by the strategies [$K$-RW, $P_C$-RW, $P(t)$-RW, and $P(t, h)$-RW ($\gamma = 16$, $H = 20$)] versus time. Here all of them consume $\mathcal{B} = 10^5$ units under varying time constraints ranging from as required by flooding to 1-RW. The coverage has been plotted in lin-log scale. The *inset* shows the plot of the coverage achieved by $P(t, h)$-RW-e, for different values of history size ($H$), under time constraint $\mathcal{T} = 200$ and bandwidth constraints 40,000 and 60,000 units

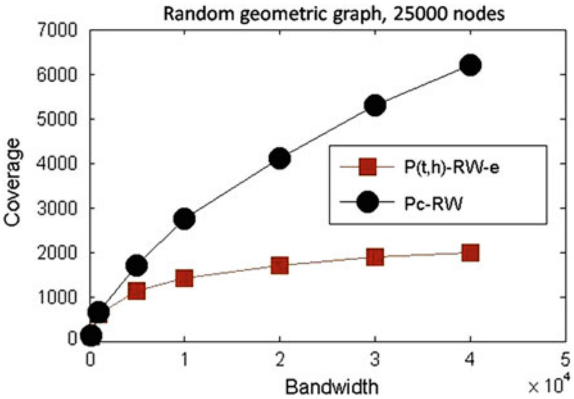


**Fig. 5.15** **a** The comparison of the coverage achieved by $P_C$-RW and $P(t, h)$-RW-e after consuming varying bandwidth upto $4 \times 10^4$, for $\mathcal{T} = 100$, in a connected random geometric graph with 25,000 nodes randomly distributed in a two-dimensional area of size 200 × 200 with the radius of communication (r) as 2 units. **b** The comparison of coverage (log scale in %)achieved by $P_C$-RW and $P(t, h)$-RW-e in the random geometric graph with the same configuration as (**a**) with the value of $r$ as 2, 3, and 4

Figure 5.15 inset shows the effect of different $H$ values on the achieved coverage in $P(t, h)$-RW-e under time constraint $\mathcal{T} = 200$ and $\mathcal{B} = 10^5$. It has been found that for a history size $H \approx 30$, the algorithm produced optimal coverage.

**Results on random geometric graphs**—Simulation results other than regular grids i.e. on 2-dimensional regular grid with diagonals connected, random geometric graphs, etc. has been be studied. A $d$-dimensional random geometric graph [32] with radius of communication $r$ is created by randomly distributing nodes in a $d$-dimensional hyperspace and connecting each pair of nodes if the Euclidian distance between them is $\leq r$. Various spatially embedded networks like sensor and ad hoc networks are often modeled as random geometric graphs. As $P(t)$-RW is not defined for graphs other than regular grids (with $d > 2$) hence, in such cases, the performance of $P(t, h)$-RW-e is compared only with $P_C$-RW. Plots in Fig. 5.15 shows that for a two-dimensional random geometric graph with 25,000 nodes and $r = 2$ units produces around (233 %) improvement in the achieved coverage by $P(t, h)$-RW-e over $P_C$-RW. For random regular graph of with $r$= 3 and 4 the performance improvements are 195 and 185 %, respectively. Further experimental evidences reveal for a highly clustered sparse network, any existing random-walk-based strategy will incur high wastage of resources due to large mutual overlap. In such cases, the proposed strategy, $P(t, h)$-RW-e, can perform significantly better than others.

## 5.6 Related Literature

The problem of estimating the coverage $C(T)$ by a 1-RW on a $d$-dimensional Euclidean space has been proposed by Dvoretzky and Erdos [11] in 1951. Indeed, the complete characterization of $C(T)$ presents a formidable mathematical challenge because the quantity is non-Markovian even when the underlying random walk is a Markov process [37]. The analytical estimate in [11] shows that during the first $T$ steps $C(T) \approx \frac{\pi \times T}{\log T}$ for $(d = 2)$, whereas $C(T) \approx T \times \gamma_d$ while $(d \geq 3)$, where $0 < \gamma_d < 1$. The qualitative difference in the results between $d \leq 2$ and higher dimensions can be explained from the remarkable observation by Pólya [31] in 1921 that a point moving randomly will, with probability 1, return infinitely often to the origin if $d \leq 2$ while if $d > 2$, then it will, again with probability 1, wander off to infinity.

Although the properties of coverage $C(T)$ due to 1-RW have been thoroughly studied in detail in general references [38], the coverage (diffusion) problem due to multiple random walkers are not solvable through simple averaging over the properties of a single random walker, even when walkers do not interact with each other. Inspite of its mathematical challenge, $C(T)$ is nevertheless used as a metric to analyze a wide range of problems in physical sciences such as diffusion-limited reaction, defect annealing, exciton trapping, etc. [22, 38] to problems related to computer networks. The recent development of experimental techniques allow the observation of events caused by single particles of an ensemble, provides additional impetus to the study of these of multi-particle diffusion problems.

In their pioneering work, Larralde et al. [21, 22] studied the dynamics of multiple $K \gg 1$ random walkers on an infinite $d$-dimensional Euclidean lattice and derived an asymptotic expression for $C(T)$, which later was followed by a more rigorous solution by Yuste et al. [41]. The most striking feature that emerges from their study is the existence of three different regimes in the dependence of $C(T)$ on $K$ and $T$.

Another direction of work on coverage related problems is focused on computing the bounds of the *cover time* ($\mathbf{C}(n)$), i.e., the expected number of steps needed for a single random walker to visit all the vertices of a finite graph of size $n$. It has been shown by Matthews [25] that for any graph $G$, $h_{min} \times H_n = C = h_{max} \times H_n$, where $h_{max}$ and $h_{min}$ are respectively the maximum and minimum of the expected number of steps taken by a random walk to move over all ordered pairs of nodes and $H_k = \ln k + \Theta(1)$ is the $k$-th harmonic number. However, the above bound is not always tight. They further reported that a random walker takes about $\frac{2^n}{\log 2^n}$ steps to visit every point. Interestingly, some threshold phenomenon is occurring at time $t = 2^{n-1} \log 2^n$, because before it some unvisited points are close to one another whereas after this time they are sparsely distributed. Feige [34] has shown that for any connected graph $G$, the cover time satisfies $(1 + o(1))n \ln n \leq \mathbf{C}(n) \leq (1 + o(1))\frac{4}{27}n^3$. For example, the lower bound is achieved for a complete graph $K_n$ and for a lollipop graph consisting of a path of length $\frac{n}{3}$ joined to a clique of size $\frac{2n}{3}$ the cover time is asymptotic to the upper bound. The cover time of a random walk on a random $r$-regular graph was studied by Cooper et al. in [8] where it was shown that with high probability (whp) for $r = 3$ the cover time is asymptotic to $\theta_r \times n \times \ln n$, where $\theta_r = \frac{(r-1)}{(r-2)}$. In a recent work [9] it has been shown that, for $K$ independent walkers each starting from $K$ different vertices on random regular graph $G$, the cover time $\mathbf{C}_G(K)$ is asymptotic to $\frac{\mathbf{C}_G}{K}$, implying exactly a linear($K$) speed-up compared to a single random walker. Alon et al. [3] computed the cover time for a wide class of graphs considering that all $K$ walkers start from the same node. Results show that a wide range of speed-ups are possible apart from linear, for example $\log K$ speed-up for a cycle and exponential in $K$ for a bar-bell graph when the walk starts at the center of bar-bell. Further, they show for $d$-dimensional lattices, hypercubes and E-R random graphs, there exists lower bound in speed-up which is linear in $K$ when $K < \mathcal{O}(\log^{1-\varepsilon} n)$, for any constant $\varepsilon > 0$. The important insight drawn is that obtaining linear speed-up using $K$-RW indeed requires bounding the value of $K$ such that mutual overlap is reduced.

More recently, coverage maximization by using multiple ($K$) walkers has become a subject of growing interest to (computer) network scientists. It may be noted that in a typical application scenario (like search), generally an object to be searched is replicated in multiple nodes, hence it is sufficient enough to cover/visit a certain fraction of nodes. From a application designer's perspective it is important to estimate the Partial Cover Time (PCT), defined as the expected number of steps required by a random walk to visit a constant $c$ fraction of the nodes, where $c$ can be 50, 80 % etc. The main analytical result is that the upper bound on the PCT is asymptotically smaller than Matthews bound [25] on the Cover Time. Intuitively, it means that on sufficiently large graphs, almost all the time used by a walk to cover the entire graph is spent trying to reach the last $\log(n)$ nodes. For a grid (a $d$-regular graph) with $d = 4$ the maximum hitting time is $n \times \log(n)$. PCT becomes $\mathcal{O}(nlog(n))$. Bisnik et al. [6] studied the performance of $K$ random walk search in terms of the replication/popularity ratio of the resource being searched for and the random walk parameters, $K$ the walker count and $T$ the TTL. They show that the random walk parameters must be a function of the resource parameters to obtain the best performance. Wu et al. [39] modeled the coverage problem of multiple random walkers initiated from $m$ randomly chosen nodes, in a random graph. They found that for small $c$ although the problem is similar to the coupon collector problem, however when $c$ is large it fails due to the finite size effect, for which they introduced a refinement. The refinement is introduced by considering the 'dirty links', i.e., the already visited links of a current node.

**Observation**—Though there is an implicit attempt to increase bandwidth utilization, none of the schemes have an explicit control over total bandwidth consumption, hence is not suitable to design bandwidth constrained coverage strategy. In the perspective of the extended coverage discussed in this chapter, it may be noted that rather than designing a strategy to achieve the speed-up in coverage, the prime focus of research is towards characterization of coverage, computing bounds of cover time and PCT, and estimation of speed-up achieved by using multiple walkers.

## 5.7 Conclusions

In this chapter we have presented the formulation of an extended coverage problem which takes into account the resource constraints in the form of consumed bandwidth $\mathcal{B}$ and latency time $\mathcal{T}$. Using methods from statistical mechanics, we have shown the design of the optimal coverage strategy $P^*(t)$-RW such that it exploits the advantages of both random walk regimes II and III by keeping the process at the regime boundary. This new algorithm yields similar efficient coverage as a single random walker, but $S = \mathcal{O}(\mathcal{B}^{(d-2)/d})$ times faster in regular grids, resulting in significant service speed-up. Alon et al. [3] has shown that for a $d$-dimensional grid, a suitably chosen $K$-RW can cover the grid with little mutual overlap within the time $\frac{T_1}{\log^{1-\varepsilon} n}$, where $T_1$ is the time taken by a single walker. Compared to that, the strategy of proliferating random walk algorithm $P^*(t)$-RW dramatically improves the result, it yields coverage with minimum overlap in much shorter time $\frac{T_1}{n^{\frac{d-2}{2}}}$.

Further, we have extended the algorithm to a class of proliferating random walk algorithms which can be used to efficiently cover the entire $\langle \mathcal{B}, \mathcal{T} \rangle$ spectrum as shown in Fig. 5.1. The derived scaling behavior of the phase boundaries can be used to estimate the effect of resource $\mathcal{B}$ and $\mathcal{T}$ preallocation in terms of obtained coverage. In other words, the minimum latency can be estimated if a certain desired level of coverage is required with a preallocated bandwidth.

The approach towards the design of the algorithm presented here for a regular grid topology can immediately be adopted to search unstructured networks with almost homogeneous node degrees e.g. sensor networks mobile ad hoc networks which are typically modeled as random geometric graphs [2, 4, 24]. However, the precise details of Eq. 5.10 might require modifications if the same strategy shall be applied to other complex networks having widely varying heterogenous nodes degrees like small world, [36], power-law [5] and the Internet. Adding small memory to nodes might help in such occasions. Perhaps some of the strategies will require replacement of the unbiased random walk by a biased random walk [13], thereby allowing the walkers to choose their next step with non-uniform probability among nearest neighbors.

In real world systems, node stores message packets using buffers of finite size, hence can hold a message queue of finite length. As a result an increase in load may lead to congestion [20, 35] of packets resulting to packet loss. However, in the optimal coverage algorithm $P^*(t)$-RW, the proliferation rate is relatively low which leads low walker density i.e. message count. Therefore, our work did not consider constrained local queue length and the corresponding problem of congestion. Here, walkers behave as isolated entities within regime III and we expect no issues of congestion with our algorithm $P^*(t)$-RW. However, in order to understand the dynamics of information dissemination, it is practically important to study the impact of congestion and congestion-aware coverage strategies are needed to be designed, especially when the time constraint $\mathcal{T}$ is much less than $T_{min}$.

In addition to our analytical approach toward todayś pre-allocation problems, our proposed coverage algorithm can also be used for upcoming sophisticated applications like service differentiation, where each node will get a different quality of service based on subscription level or its history of cooperation [24, 26].

## References

1. Admic, L.A., Lukose, R.M., Puniyani, A.R., Huberman, B.A.: Search in power-law networks. Phys. Rev. E **64**(4), 046135 (2001)
2. Ahn, J., Kapadia, S., Pattem, S., Sridharan, A., Zuniga, M., Jun, J., Avin, C., Krishnamachari, B.: Empirical evaluation of querying mechanisms for unstructured wireless sensor networks. ACM SIGCOMM Comput. Commun. Rev. **38**(3), 17–26 (2008)
3. Alon, N., Avin, C., Koucky, M., Kozma, G., Lotker, Z., Tuttle, M.R.: Many random walkers are faster than one. In: Proceedings of 20th ACM Annual Symposium (SPAA) (2008)
4. Avin, C., Brito, C.: Efficient and robust query processing in dynamic environments using random walk techniques. In: Proceedings of 3rd ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN'04) (2004)
5. Barabasi, A.: Scale-free networks. Sci. Am. **288**(2), 60–69 (2003)
6. Bisnik, N., Abouzeid, A.A.: Optimizing random walk search algorithms in p2p networks. Comput. Netw. **51**(2), 1499–1514 (2007)
7. Clarke, I., Sandberg, O., Wiley, B., Hong, T.W.: Freenet: a distributed anonymous information storage and retrieval syste. In: Proceedings of of International Workshop on Designing Privacy Enhancing Technologies: Design Issues in Anonymity and Unobservability, pp. 46–66 (2001)
8. Cooper, C., Frieze, A.: The over time of random regular graphs. SIAM J. Discret. Math. **18**(4), 728–740 (2005)
9. Cooper, C., Frieze, A., Radzik, T.: Multiple random walks in random regular graphs. SIAM J. Discret. Math. **23**(4), 1738–1764 (2009)
10. Dhillon, S.S., Mieghem, P.V.: Searching with multiple random walk queries. In: Proceedings of 18th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'07) (2007)
11. Dvoretzky, A., Erdos, P.: Some problems on random walk in space. In: Proceedings of the 2nd Berkley Symposium on Mathematical Statistics and Probability. University of California Press, Berkley (1951)
12. Fall, K.: A delay tolerant network architecture for challenged internets. In: Proceedings of ACM SIGCOMM, pp. 27–34 (2003)
13. Fronczak, A., Fronczak, P.: Biased random walks in complex networks: the role of local navigation rules. Phys. Rev. E **80**, 065164 (2009)
14. Gnutella, R.F.C.: The gnutella protocol specification v0.4. Technical report (2004)
15. Guclu, H., Yuksel, M.: Method to find community structures based on information centrality. IEEE Trans. Parallel Distrib. Syst. **20**(5), 667–679 (2009)
16. Heinzelman, W.R., Kulik, J., Balakrishnan, H.: Adaptive protocols for information dissemination in wireless sensor network. In: Proceedings of 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking, MobiCom 99, pp. 174–185 (1999)
17. Khelil, A., Becker, C., Tian, J., Rothermel, K.: An epidemic model for information diffusion in manet. In: Proceedings of 5th ACM International Workshop on Modeling Analysis and Simulation of Wireless and Mobile System (MSWiM 02), pp. 54–60 (2002)
18. Król, D.: On modelling social propagation phenomenon. In: Nguyen, N.T., Attachoo, B., Trawiński, B., Somboonviwat, K. (eds.) Intelligent Information and Database Systems, Lecture Notes in Computer Science, pp. 227–236. Springer Publishing (2014)
19. Król, D.: Propagation phenomenon in complex networks: theory and practice. New Gener. Comput. **32**(3–4), 187–192 (2014)

20. Kwong, K.W., Tsang, D.H.K.: A congestion-aware search protocol for heterogeneous peer-to-peer networks. J. Supercomputing **36**(3), 265–282 (2006)
21. Larralde, H., Trunfio, P., Havlin, S., Stanley, H.E., Weiss, G.H.: Number of distinct sites visited by n random walkers. Phys. Rev. A **45**(10), 7128–7138 (1992)
22. Larralde, H., Trunfio, P., Havlin, S., Stanley, H.E., Weiss, G.H.: Territory covered by n diffusing particles. Nature **355**, 423–426 (2002)
23. Liang, J., Kumar, R., Ross, K.W.: Understanding kazaa. In: Proceedings of 19th IEEE Annual Computer Communications Workshop (2004)
24. Mabrouki, I., Lagrange, X., Froc, G.: Random walk based routing protocol for wireless sensor networks. In: Proceedings of 2nd International Conference on Performance Evaluation Methodologies and Tools (Inter-Perf '07) (2007)
25. Matthews, P.: Annals of Probability (1998)
26. Mekouar, L., Boutaba, R.: A contribution-based service differentiation scheme for peer-to-peer systems. J. Peer-to-Peer Netw. Appl. **2**, 146–163 (2009)
27. Nandi, S., Brusch, L., Deutsch, A., Ganguly, N.: Coverage-maximization in networks under resource constraints. Phys. Rev. E **81**, 0611241–0611246 (2010)
28. Nandi, S., Brusch, L., Ganguly, N.: Coverage maximization in small world network under bandwidth constraint. ACM SIGCOMM (poster) (2008)
29. Nandi, S., Pal, A., Ganguly, N.: When and how much random walkers should proliferate for a fast and efficient walk? In: Proceedings of Centenary Conference on Managing Complexity in a Distributed System (MCDES) (2008)
30. Oikonomou, K., Kogias, D., Stavrakakis, I.: A study of information dissemination under multiple random walkers and replication mechanisms. In: Proceedings of ACM MobiOpp (2010)
31. Pólya, G.: Über eine aufgabe der wahrscheinlichkeitsrechnung betreffend die irrfahrt im strassennetz. Mathematische Annalen **84**(1–2), 149–160 (1921)
32. Saha, S., Ganguly, N.: Coverage maximization under resource constraints using a nonuniform proliferating random walk. Phys. Rev. E **87**, 022, 807 (2013)
33. Stauffer, A.O., Barbosa, V.C.: Probabilistic heuristics for disseminating information in networks. IEEE/ACM Trans. Netw. **15**(2), 425–435 (2007)
34. Feigei, U.: Random structures and algorithms. Comput. Commun. Rev. (1995)
35. Wang, D., Jing, Y., Zhang, S.: Traffic dynamics based on a traffic awareness routing strategy on scale-free networks. Physica A **387**, 3001–3007 (2008)
36. Watts, D.J., Strogatz, S.H.: Collective dynamics of small-world networks. Nature **393**, 440–442 (1998)
37. Weiss, G., Dayan, I., Kiefer, S.H.J.E., Stanley, H.L.H.E., Turnfio, P.: Collective dynamics of small-world networks. Physica A (1992)
38. Weiss, G.H.: Aspects and Applications of Random Walk. North-Holland, Amsterdam (1994)
39. Wu, B., Kshemkalyani, A.D.: Modeling message propagation in random graph networks. Comput. Commun. **31**, 4138–4148 (2008)
40. Wu, J.J., Gao, Z.Y., Sun, H.J., Huang, H.J.: Congestion in different topologies of traffic networks. Europhys. Lett. 74(3) (2006)
41. Yuste, S.B., Acedo, L.: Number of distinct sites visited by n random walkers on an euclidean lattice. Phys. Rev. E **61**(3), 2340–2347 (2000)

# Chapter 6
# Petri Net-Based Modelling and Simulation of Transport Network Segments

**František Čapkovič**

**Abstract** Three kinds of Petri nets are utilized here in order to model and simulate segments of a transport network. The segments are understood to be agents. A suitable cooperation of the agents makes it possible to model and simulate the vehicle flow propagation in the network. Place/transitions Petri nets (P/T PN) are utilized in order to find the safe and unambiguous structure of the controller for the traffic lights placed at the road intersections. After finding such a structure the time specifications are assigned to the P/T PN. Thus, timed Petri nets (TPN) arise from P/T PN. The TPN model yields the possibility to analyze the time relations among the traffic lights. Subsequently, hybrid Petri nets, more precisely first-order hybrid Petri nets, are used for finding the flows of vehicles moving on the roads within the bounds of possibility determined by the traffic lights. A generalization towards the more complicated segment is pointed out too. A possibility of the modular interconnection of the segments is mentioned in connection with the vehicle flow propagation in the transport network.

## 6.1 Introduction and Preliminaries

Transport networks are a specific kind of the real world networks. They strongly affect present life as well as the human social behaviour, living environment, industry (e.g. in case of the just-in-time manufacturing), and many other areas. On the other hand they bring many problems that need to be solved. The vehicle flow propagation in the transport networks is one of the examples of the propagation phenomenon in complex networks in general [19].

It is the everyday's race against time. Consequently, the most important task is to assure the transport safety. Therefore, modelling and simulation are important parts of the transport systems design, control and continuous running. There are many approaches how to deal with them. General understanding the problems and the formulation of corresponding approaches that describe how to solve them are

F. Čapkovič (✉)
Institute of Informatics, Slovak Academy of Sciences, Dúbravská cesta 9,
845 07 Bratislava, Slovakia
e-mail: Frantisek.Capkovic@savba.sk

D. Król et al. (eds.), *Propagation Phenomena in Real World Networks*,
Intelligent Systems Reference Library 85, DOI 10.1007/978-3-319-15916-4_6

introduced e.g. in [1–3, 11, 12, 14, 15, 22]. These approaches do not use PN. The PN-based approaches (alternative to the previous ones) can be found e.g. in [17, 18, 20, 28–30] and in many other papers. Here, in this chapter, the agent-based approach to Petri net-based modelling and simulation of transport systems is presented. Motivation for this follows especially from: (i) the ability of PN to yield both the graphical model and the analytical one; (ii) the possibility to use the existing methods for analyzing the PN structure and finding the PN basic properties like reachability of states, invariants, etc.; (iii) the ability to remove emerging deadlocks; (iv) the possibility to work with simpler modules and to synthesize more complicated structures from the modules; (v) the ability to synthesize supervisors in order to control the PN-based models; (vi) the fact that no paper from the quoted PN-based ones tries to solve these problems simultaneously.

Three kinds of PN are utilized here in order to model and simulate segments (modules) of the transport network. Namely, place/transition Petri nets (P/T PN), timed Petri nets (TPN) and hybrid Petri nets are used. While P/T PN and TPN are used for modelling and simulation of discrete modules representing the traffic lights placed at the road intersections, HPN are used for modelling and simulation of the road network together with the flows of vehicles. P/T PN handle the discrete tokens representing the step-by-step evolution of the states driven by discrete events. P/T PN work without any timing. TPN yield discrete time functions specifying the duration of the states. HPN offer the possibility to model the flows of vehicles controlled by the discrete events occurring in P/T PN and/or TPN.

P/T PN [21, 23] are bipartite directed graphs with two kinds of nodes and two kinds of edges. While the nodes are represented by the places and transitions, the edges are represented by the arcs directed from places to transitions and the arcs directed in opposite direction. Thus, P/T PN is a triple $PN = \langle P, T, B\rangle$, where $P$, $|P| = n$, is a finite set of places and $T$, $|T| = m$ is a finite set of transitions. They have to satisfy the conditions $P \cup T \neq \emptyset$;   $P \cap T = \emptyset$;   $B \subseteq (P \times T) \cup (T \times P)$ Here, $F \subseteq (P \times T)$ represents the set of the arcs directed from places to transitions, while $G \subseteq (T \times P)$ expresses the set of the arcs directed from transitions to places. The set of input and output transitions of a place $p \in P$ are, respectively, denoted by ${}^{\bullet}p$ and $p^{\bullet}$. Similarly, the set of input and output places of a transition $t \in T$ are, respectively, denoted by ${}^{\bullet}t$ and $t^{\bullet}$.

However, there is also a possibility to evolve the marking of the places in P/T PN. A function $M_0 : P \to \mathbb{N}_0$ is the initial marking [represented below in (6.1) by the initial state vector $\mathbf{x}_0$], where $\mathbb{N}_0$ is the set of nonnegative integers. A transition $t \in T$ is said to be enabled at $M_0$ if, for all $p \in {}^{\bullet}t$, $M_0(p) \geq 1$. A transition may be fired if it is enabled. Firing a transition $t$ at marking $M$ removes one token from each of its input places and puts one token to each of its output places. It leads to a new marking $M'$. This process is denoted by $M[t > M'$, however it can be represented by the discrete state equation (6.1) introduced below. The set of all markings reachable from $M_0$ is denoted by $R(M_0)$. The notation *set of feasible states* is sometimes used too. The PN marking represents step-by-step evolution of the PN *dynamics*. The dynamics can be expressed formally by the quadruplet $\langle X, U, \delta, \mathbf{x}_0\rangle$, $X \cap U = \emptyset$, where $X$ is a set of state (marking) vectors $\mathbf{x}_k$ expressing by its entries the states (0—no token, 1—one token, 2—two tokens,...) of elementary places in different steps $k = 0, 1, \ldots, K$ of the dynamics development, $U$ is a set of control vectors $\mathbf{u}_k$—i.e. state vectors of transitions—expressing by its entries the states (1—enabled, 0—disabled) of elementary transitions in different steps $k = 0, 1, \ldots, K$. $\delta : X \times U \to X$ is the P/T PN transition function and $\mathbf{x}_0$ is the initial state (marking). The sets $B$, $F$, $G$ can be expressed, respectively, by the incidence matrices $\mathbf{B}$, $\mathbf{F}$, $\mathbf{G}$, where $\mathbf{B} = \mathbf{G}^T - \mathbf{F}$. Consequently, the PN dynamics (represented formally by the PN transition function $\delta$) can be expressed by the restricted linear discrete system

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{B} \cdot \mathbf{u}_k \tag{6.1}$$

$$\mathbf{F} \cdot \mathbf{u}_k \leq \mathbf{x}_k; \quad k = 0, 1, \ldots, K \tag{6.2}$$

The nonzero entries of the matrices $\mathbf{F}$, $\mathbf{G}$ represent, respectively, the existence and/or multiplicity of corresponding edges (arcs) between places and transitions and vice versa.

A transition $t$ is said to be live if, for any marking $M \in R(M_0)$, there exists a sequence of transitions firable from $M$ which contains $t$. A PN is said to be live if all the transitions are live. A PN is said to be safe if for any marking $M \in R(M_0)$, $M(p_i) \leq 1$, for all places $p_i \in P$.

P/T PN do not contain time parameters. The time parameters can be introduced into places, transitions, arcs, even into tokens representing the marking [24, 25]. Here, in this chapter we will use only the time parameters assigned to the P/T PN transitions. Thus, the TPN arise from P/T PN.

HPN [13] are a combination of discrete PN (P/T PN or TPN) and continuous PN. They model the coexistence of discrete and continuous variables and have two groups of places and transitions—discrete and continuous. Consequently, there are three kinds of directed arcs here: (i) between discrete places and discrete transitions; (ii) between continuous places and continuous transitions; (iii) between discrete places and continuous transitions as well as between the continuous places and discrete transitions. The discrete places and transitions handle discrete tokens. The continuous places and transitions handle continuous variables. Here, they can model the flows of vehicles. The set of places $P = P_d \cup P_c$, where $P_d$ is a set of discrete places and $P_c$ is a set of continuous places (figured by double concentric circles). The set of transitions $T = T_d \cup T_c$, where $T_d$ is a set of discrete transitions and $T_c$ is a set of continuous transitions (figured by double rectangles). $T_d$ contains a subset of immediate (no-timed) transitions and/or a subset of timed transitions (deterministic and/or non-deterministic).

Firstly, the P/T PN model of the traffic lights cooperation will be introduced in this chapter. Possible ambiguities occurring in such a model are removed by means of a supervisor which can be synthesized by means of the methods used in control theory for discrete-event systems (DES). Then, the time parameters will be included into the transitions of the P/T PN model. In this manner, the TPN model is obtained. Next, the HPN model will be created by means of connecting the TPN model of traffic lights with the continuous PN model of the intersection of two roads. After this, the possibility of the generalization for bidirectional running the vehicles on both intersecting roads will be mentioned. Namely, the HPN model of the bidirectional running of vehicles on the roads will be created and a simple multiplexer of the corresponding traffic lights will be introduced. For all simulations based on the mentioned models the Matlab tool HYPENS [26, 27] will be used. Simultaneously, author's experience with using this tool [5, 8–10] as well as with the modular agent-based approach to modelling and supervisory control of complex systems [4, 7] will be applied.

The segments (modules) of the transport network are understood here to be agents. Then, the suitable agent cooperation makes possible to model and simulate the vehicle flow propagation in the larger segments or in the whole network. The first segment (module, agent)—the safe and unambiguous structure of the controller for the traffic lights placed at the road intersections—is modelled by TPN that arose from P/T PN after assigning time specifications to their transitions. The second segment—used for finding the flows of vehicles moving on the roads within the bounds of possibility determined by the traffic lights—is modelled by HPN. The third segment—the multiplexer of the traffic lights belonging to the intersecting roads—is modelled by TPN.

## 6.2 The Agent-Based Approach to the Traffic Lights Control

Traffic lights are one of the most important parts of a transport system. As it is presented in [20], the safety control of the traffic lights is not any simple thing. However, due to the importance of the traffic lights, new approaches to their control are continuously explored. Moreover, there exist many consecutive intersections in large-scale transport systems. Therefore, a modular approach to modelling and control of the traffic lights seems to be convenient. To illustrate the complexity of a simple module, consider the traffic light $L_i$ belonging to the road $D_i$ at an intersection. The simplest form of the intersection is displayed below in Fig. 6.3. The traffic lights can be modelled by P/T PN given in Fig. 6.1. Here, $A_i^1$ and $A_i^2$ can be understood to be the cooperating agents.

The interpretation of the PN places and transitions is the following: $p_1$—the green color is on; $p_2$—the end of the green; $p_3$—the yellow color is on; $p_4$—the end of the yellow; $p_5$—the red color is on; $p_6$—the start of the green; $p_7$—the light is green responsive; $p_8$—the start of the yellow; $p_9$—the light is yellow responsive; $p_{10}$—the start of the red; $p_{11}$—the conversion to the yellow; $p_{12}$—the conversion to the red.

The transitions $t_j$, $j = 1, \dots, 8$, represent the discrete events—e.g. the starts and/or ends of the individual colors.

In the right picture in Fig. 6.1 where are two traffic lights the numbering of the places and transitions of the couple of agents continues—$p_{13}$ corresponds with $p_1$, $p_{14}$ corresponds with $p_2$, etc. It is also necessary to point out the *feedback* from $p_5$

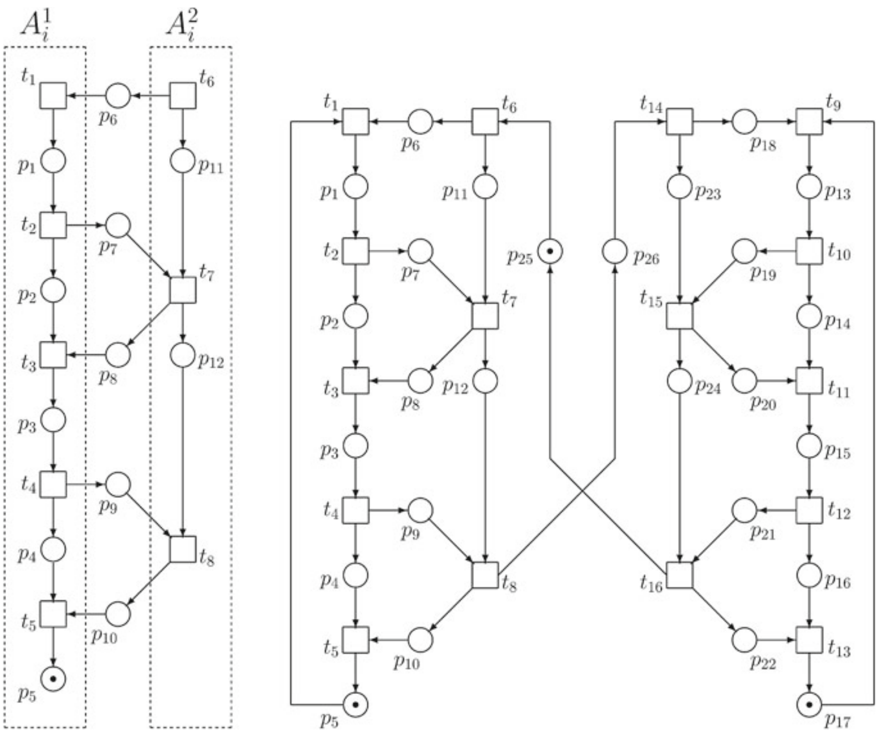


**Fig. 6.1** The agent-based P/T PN model of the single traffic light (the *left picture*) and the P/T PN model of two cooperating traffic lights, when each of them is placed on one of the intersecting roads (the *right picture*)

to $t_1$ as well as the *feedback* from $p_{17}$ to $t_9$. These feedbacks realize the cooperation of the traffic lights. Namely, $p_{25}$ and $p_{26}$ make possible mutual switching the colors of the traffic lights $L_1$ and $L_2$.

The modular approach to solving the problem of traffic light control is introduced in [28, 29]. However, the reachability graph (RG) of such a structure is ambiguous. Namely, the succession of the control steps is not straight-lined. There exists branching in the RG. To remove the ambiguity, it is necessary to add a supervisor which will guarantee the RG without any branching—i.e. the single succession of the states. Using the PN-based supervision theory of discrete event systems [6, 16] a supervisor can be synthesized.

### *6.2.1 The Supervisor Synthesis*

The supervisor synthesis is performed by means of the mutual exclusion of the controversial states. The exclusion is based on the P/T PN place invariants (P-invariants).

Namely, the P-invariant of the P/T PN is the vector $\mathbf{v}$ satisfying the condition

$$\mathbf{v}^T \cdot \mathbf{x} = \mathbf{v}^T \cdot \mathbf{x}_0 \tag{6.3}$$

for each state vector $\mathbf{x}$ reachable from the initial state vector $\mathbf{x}_0$. Alternative definition is used in the form

$$\mathbf{v}^T \cdot \mathbf{B} = \mathbf{0} \tag{6.4}$$

and in case of more P-invariants it has the following form

$$\mathbf{V}^T \cdot \mathbf{B} = \mathbf{0} \tag{6.5}$$

with $\mathbf{V}$ being the matrix. The columns of $\mathbf{V}$ contain the P-invariants which we are looking for. Imposing the conditions on the linear combinations of the state vectors entries in the form as follows

$$\mathbf{L}_p \cdot \mathbf{x} \leq \mathbf{b} \tag{6.6}$$

and removing the inequality by introducing the $(n_s \times 1)$ vector $\mathbf{x}_s$ of *slack* variables we have

$$\mathbf{L}_p \cdot \mathbf{x} + \mathbf{x}_s = \mathbf{b} \tag{6.7}$$

and for the initial state

$$\mathbf{L}_p \cdot \mathbf{x}_0 + \mathbf{x}_s^0 = \mathbf{b} \tag{6.8}$$

where $\mathbf{L}_p$ is $(n_s \times n)$ matrix of integers and $\mathbf{b}$ is $(n_s \times 1)$ vector of integers. Thus, the vector $\mathbf{x}_s$, being the supervisor state vector, can be obtained. The vector $\mathbf{x}_s^0$ represents the initial state vector of the supervisor. Comparing (6.7) with (6.5) we can write

$$(\mathbf{L}_p \ \mathbf{I}_s) \cdot (\mathbf{B}^T \ \mathbf{B}_s^T)^T = \mathbf{0} \tag{6.9}$$

and after multiplying

$$\mathbf{B}_s = -\mathbf{L}_p \cdot \mathbf{B} \tag{6.10}$$

where $\mathbf{I}_s$ is $(s \times s)$ identity matrix, $\mathbf{B}_s$ represents the structure of the supervisor, $(\mathbf{L}_p \ \ \mathbf{I}_s)$ corresponds to (more precisely, it is forced instead) $\mathbf{V}^T$ and $(\mathbf{B}^T \ \ \mathbf{B}_s^T)^T$ corresponds to (more precisely, it is forced instead) $\mathbf{B}$. Because $\mathbf{B}_s = \mathbf{G}_s^T - \mathbf{F}_s$, the supervisor incidence matrices $\mathbf{F}_s$, $\mathbf{G}_s$ can be easily found.

In more general cases it is necessary to impose conditions also on the vector of transitions and/or on the Parikh's vector $\mathbf{v}_P = \sum_{k=0}^{K} \mathbf{u}_k$ (relative to the step-by-step evolution of the system (6.1)) in the form

$$\mathbf{L}_p \cdot \mathbf{x} + \mathbf{L}_t \cdot \mathbf{u} + \mathbf{L}_{v_P} \cdot \mathbf{v}_P \leq \mathbf{b} \tag{6.11}$$

where $\mathbf{L}_t$, $\mathbf{L}_{v_P}$ are $(n_s \times m)$ matrices of integers. The Parikh's vector $\mathbf{v}_P$ is coherent with the evolution of the P/T PN marking from the initial state $\mathbf{x}_0$ into the terminal state $\mathbf{x}_k$. To set priorities among firing the P/T PN transitions, the Parikh's vector is decisive (authoritative). Therefore, only the simplified condition (6.11) in the form

$$\mathbf{L}_{v_P} \cdot \mathbf{v}_P \leq \mathbf{b} \tag{6.12}$$

has to be imposed. Then, the incidence matrices of the supervisor and the initial state of the supervisor are the following

$$\mathbf{F}_s = \max(\mathbf{0},\ \mathbf{L}_{v_P}) \tag{6.13}$$
$$\mathbf{G}_s^T = \max(\mathbf{0},\ (-\max(\mathbf{0},\ (\mathbf{L}_{v_P})))) - \min(\mathbf{0},\ (\mathbf{L}_{v_P})) \tag{6.14}$$
$$\mathbf{x}_s^0 = \mathbf{b} - \mathbf{L}_{v_P} \cdot \mathbf{v}_P^0 \tag{6.15}$$

Using this methodology in our case, it is possible to ensure the priorities $\lambda(t)$ at firing the transitions as follows

$$\lambda(t_5) > \lambda(t_{14}) \tag{6.16}$$
$$\lambda(t_{13}) > \lambda(t_6) \tag{6.17}$$

It means that

$$-\lambda(t_5) + \lambda(t_{14}) < 0 \tag{6.18}$$
$$\lambda(t_6) - \lambda(t_{13}) < 0 \tag{6.19}$$

Consequently, the supervisor was synthesized by means of substituting the matrix

$$\mathbf{L}_{v_P} = \begin{pmatrix} 0&0&0&0&-1&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&1&0&0&0&0&0&0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0 \end{pmatrix} \tag{6.20}$$

into (6.12). Thus, from (6.13), (6.14) we have

$$\mathbf{F}_s = \begin{pmatrix} 0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \end{pmatrix} \tag{6.21}$$

$$\mathbf{G}_s^T = \begin{pmatrix} 0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0 \end{pmatrix} \tag{6.22}$$

The P/T PN structure of the supervised system is displayed in Fig. 6.2. The supervisor is created by the P/T PN places $p_{27}$, $p_{28}$. With respect to the incidence matrices $\mathbf{F}_s$, $\mathbf{G}_s$, the directed arcs from $p_{27}$ to $t_{14}$ and from $p_{28}$ to $t_6$ as well as from $t_5$ to $p_{27}$ and from $t_{13}$ to $p_{28}$ incorporate the supervisor into the original (non-supervised) P/T PN displayed in Fig. 6.1 left. Thus the unambiguous reachability tree (RT) can be found.

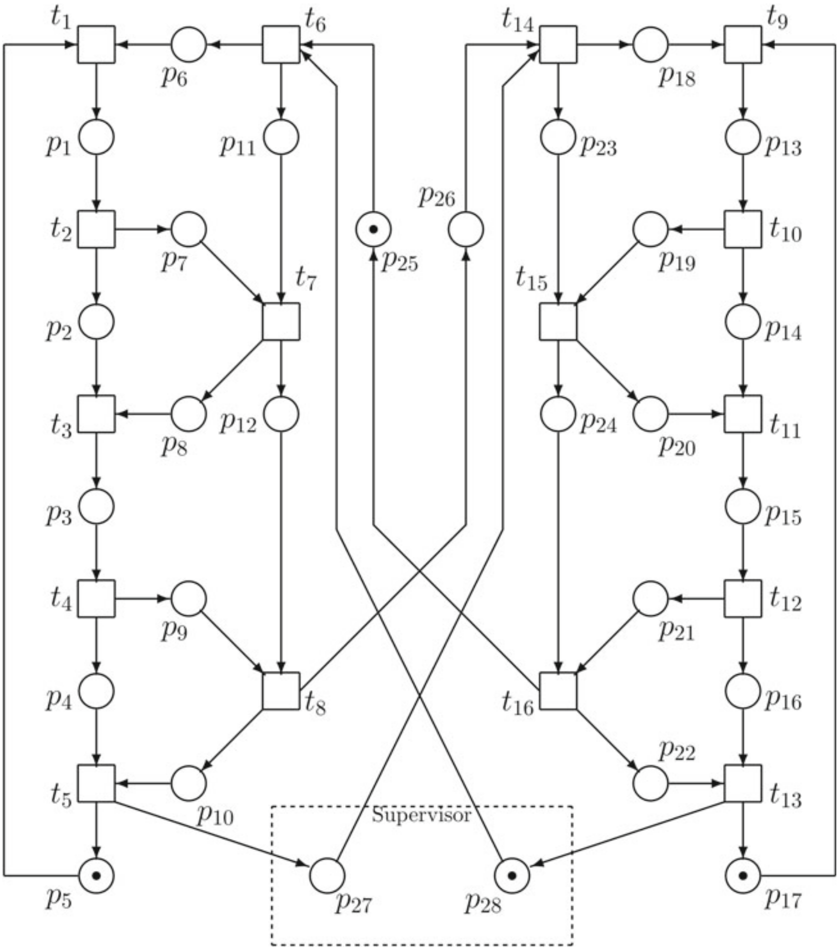


**Fig. 6.2** The agent-based P/T PN model of the supervised traffic lights on the intersection of two roads

The reachability tree (RT) yields all states $R(M_0)$ reachable from the initial state $\mathbf{x}_0$. Connecting all leaves of the RT with the same name into one we obtain the reachability graph (RG). The complexity of computations depends on the RT size and complexity of its structure. The adjacency matrix for RT and the adjacency matrix for the corresponding RG is the same quadratic matrix. The dimensionality of the adjacency matrix corresponds to the number of the RT nodes. The nodes represent the P/T PN state vectors. The RT root $N_1$ represents the initial state vector $\mathbf{x}_0$. The RT leaves are the state vectors reachable from the initial state. The state vectors can be stored as the columns of the matrix $\mathbf{X}_{reach}$. The ideal RT is the RT without any branching. It means that the immediate succession of states—i.e. the structure where the nodes lie on a straight line—is the most favourable case. After introducing the supervisor, we have obtained just the simplest RT and RG. Namely, the unambiguous RT without any branching corresponding to the supervised P/T PN model given in Fig. 6.2 has the following form

$$\begin{aligned}&N_1 \xrightarrow{t_6} N_2 \xrightarrow{t_1} N_3 \xrightarrow{t_2} N_4 \xrightarrow{t_7} N_5 \xrightarrow{t_3} N_6 \xrightarrow{t_4} N_7 \xrightarrow{t_8} N_8 \xrightarrow{t_{14}} N_9 \xrightarrow{t_9} N_{10} \xrightarrow{t_{10}} \\ &N_{11} \xrightarrow{t_5} N_{12} \xrightarrow{t_{11}} N_{13} \xrightarrow{t_{12}} N_{14} \xrightarrow{t_{16}} N_{15} \xrightarrow{t_{13}} N_1 .\end{aligned} \tag{6.23}$$

Here, the nodes $N_k$, $k = 1, \dots, 16$, represent the P/T PN model state vectors $\mathbf{x}_0, \dots, \mathbf{x}_{15}$ being the rows of the following matrix $\mathbf{X}^T_{\text{reach}}$. In each row the last two entries concern the supervisor state.

$$\mathbf{X}^T_{\text{reach}} = \left(\begin{array}{cccccccccccccccccccccccccccc}
0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&1&0&0&1\\
0&0&0&0&1&1&0&0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0\\
1&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0\\
0&1&0&0&0&0&1&0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0\\
0&1&0&0&0&0&0&1&0&0&0&1&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0\\
0&0&1&0&0&0&0&0&0&0&0&1&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&1&0&0&0&0&1&0&0&1&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&1&0&0\\
0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&1&1&0\\
0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&1&1&0&0&0&0&1&0&0&0&0&0\\
0&0&0&0&1&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0\\
0&0&0&0&1&0&0&0&0&0&0&0&0&1&0&0&0&0&1&0&0&0&1&0&0&0&0&0\\
0&0&0&0&1&0&0&0&0&0&0&0&0&1&0&0&0&0&0&1&0&0&0&1&0&0&0&0\\
0&0&0&0&1&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&1&0&0&0&0\\
0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&1&0&0&1&0&0&0&0\\
0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&1&0&0&1&0&0&0
\end{array}\right) \tag{6.24}$$

In our case the RG certifies that the evolution of the supervised P/T PN model is unambiguous, correct and sound. However, the time relations are most important for the traffic lights placed at the road intersections. Consequently, let us introduce the time into the transitions of the supervised P/T PN model given in Fig. 6.2.

### 6.2.2 *Timed Petri Net-Based Model*

Namely, for example, let the red color on the traffic light shines for 30 time units (seconds), while the green color shines for 27 units and the amber (yellow) color shines for 3 units. Thus, the times assigned to the particular transitions of the P/T PN-based model are the following

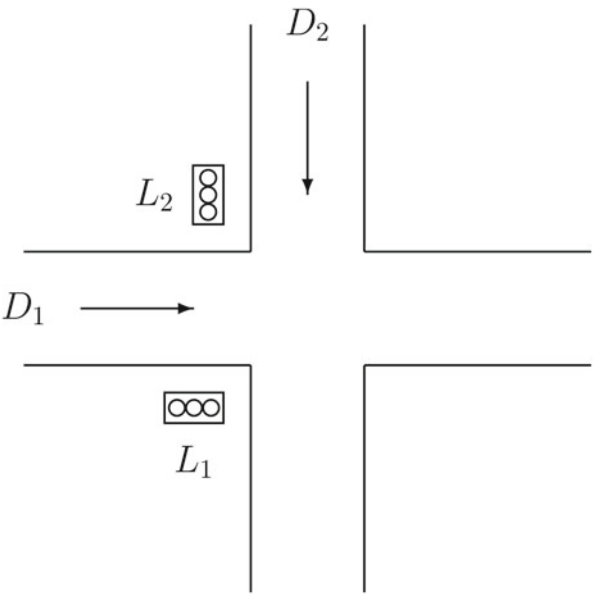


**Fig. 6.3** The simple intersection. $D_1$, $D_2$ represent the directions of traffic on the corresponding roads, while $L_1$, $L_2$ represent the corresponding traffic lights

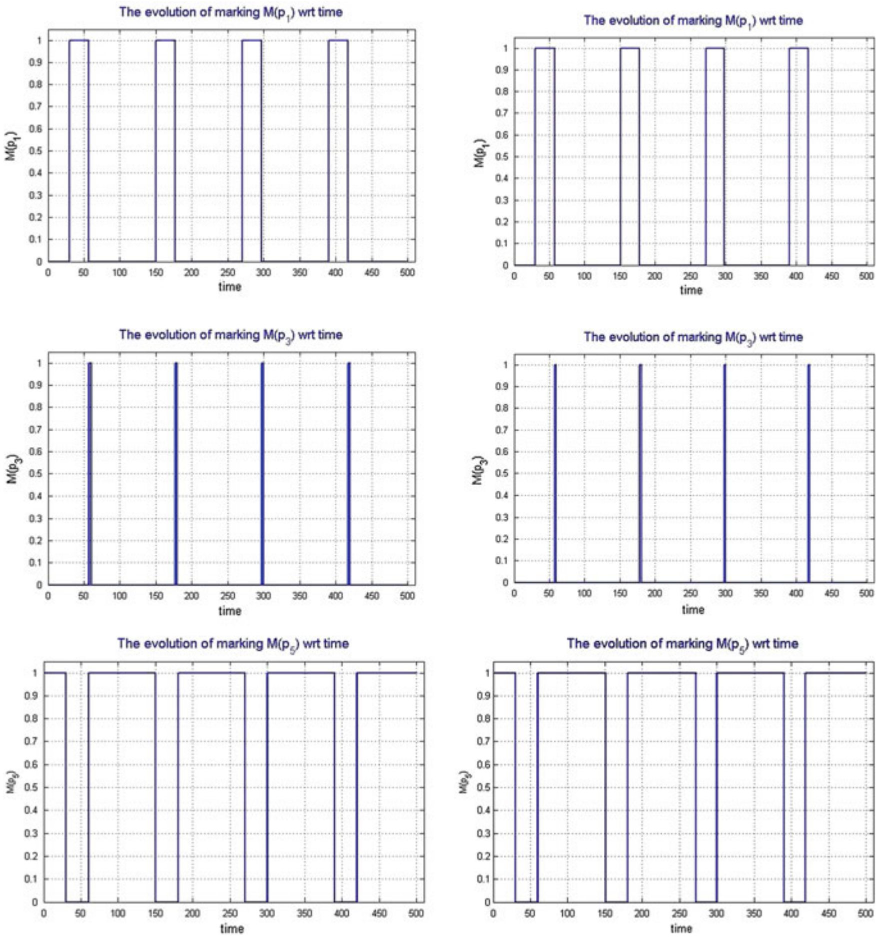
$$\begin{aligned}&t_1 \triangleq 30,\ t_2 \triangleq 27,\ t_3 \triangleq 0.01,\ t_4 \triangleq 3,\ t_5 \triangleq 0.01,\ t_6 \triangleq 0.01,\\ &t_7 \triangleq 0.01,\ t_8 \triangleq 0.01,\ t_9 \triangleq 30,\ t_{10} \triangleq 27,\ t_{11} \triangleq 0.01,\ t_{12} \triangleq 3,\\ &t_{13} \triangleq 0.01,\ t_{14} \triangleq 0.01,\ t_{15} \triangleq 0.01,\ t_{16} \triangleq 0.01.\end{aligned} \tag{6.25}$$

In this manner we obtained the TPN-based model.

Consider now crossing of two simple roads given in Fig. 6.3. Apply now timing the colors on the traffic lights for the whole intersection. For simulation using the TPN model we will use the tool HYPENS in Matlab. In case of the deterministic timing of the TPN transitions, the simulation results are given in the left column of pictures introduced in Fig. 6.4 (for the traffic light L1) and in the left column of pictures introduced in Fig. 6.5 (for the traffic light L2).

In order to test a possibility of changing the deterministic timing of the TPN transitions (i.e. to shorten or enlarge the time delays) in a small range, we can use a non-deterministic timing with different kinds of probability distribution (discrete uniform, exponential, Poisson's, Rayleigh's, Weitbull's, etc.). When the achieved results do not differ distinctly from the results corresponding to the original deterministic timing, changing the time delays is not necessary.

Consider now the non-deterministic timing of the transitions with the discrete uniform probability distribution $f_x = 1/(b-a)$ when $x \in (a, b)$ and $f_x = 0$ otherwise, with $a$, $b$ for individual transitions being the entries of the vectors


The evolution of marking M(p$_1$) wrt time

The evolution of marking M(p$_1$) wrt time

The evolution of marking M(p$_3$) wrt time

The evolution of marking M(p$_3$) wrt time

The evolution of marking M(p$_5$) wrt time

The evolution of marking M(p$_5$) wrt time

**Fig. 6.4** The evolution of markings of the places $p_1$, $p_3$, $p_5$ in time representing, respectively, the *green*, *yellow* and *red colors* in the first traffic light $L_1$ are displayed. The simulation results at the deterministic timing the TPN transitions are placed in the *left column*, while the simulation results at the non-deterministic timing the TPN transitions (with the discrete uniform probability distribution) are displayed in the *right column*

$$\mathbf{a} = (29, 26, 0.005, 2, 0.005, 0.005, 0.005, 0.005, 29, 26, 0.005, 2, 0.005, 0.005, 0.005, 0.005)$$
$$\mathbf{b} = (31, 28, 0.015, 4, 0.015, 0.015, 0.015, 0.015, 31, 28, 0.015, 4, 0.015, 0.015, 0.015, 0.015)$$

Then, we obtain the simulation results given in the right column of pictures introduced in Fig. 6.4 (for the first traffic light $L_1$) and in the right column of pictures introduced in Fig. 6.5 (for the second traffic light $L_2$).

As we can see (cf. the left and right columns of pictures in Fig. 6.4 as well as the left and right columns of pictures in Fig. 6.5), the results are practically the same in both the deterministic case and non-deterministic one. It means that in

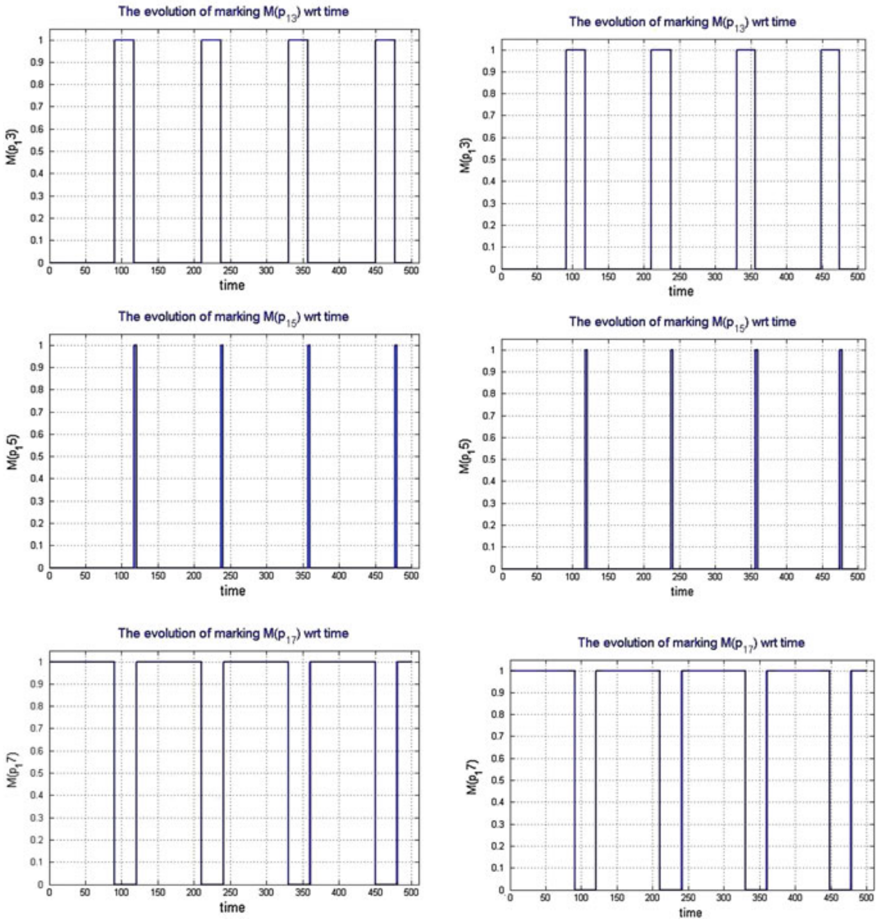


**Fig. 6.5** The evolution of markings of the places $p_{13},\ p_{15},\ p_{17}$ in time representing, respectively, the *green*, *yellow* and *red colors* in the second traffic light $L_2$ are displayed. The simulation results at the deterministic timing the TPN transitions are placed in the *left column*, while the simulation results at the non-deterministic timing the TPN transitions (with the discrete uniform probability distribution) are displayed in the *right column*

our case the control of the traffic lights seems to be robust. Therefore, it is useless to change the deterministic timing (i.e. the time delays of the TPN transitions). However, unfortunately, we cannot generalize from the only one case. In case of other parameters of the discrete uniform probability distribution or in case of using a different kind of the probability distribution (e.g. exponential, Poisson's, Rayleigh's, Weitbull, etc.) the situation can be completely different.

## 6.3 Modelling the Intersection by Means of Hybrid Petri Nets

The real intersection given in Fig. 6.3 can be modelled by means of HPN. The continuous PN given in Fig. 6.6 models the crossing roads. The directed arcs to/from $T_2$ and to/from $T_6$ point out the interconnections with the TPN model corresponding to the P/T PN model given in Fig. 6.2. During simulation the HPN model is able to offer the corresponding flows of the vehicles which are moving on the roads. The above introduced discrete TPN (corresponding to P/T PN displayed in Fig. 6.2) models switching the colors of the traffic lights. The edges leading to/from the discrete places, which are the components of the TPN model, denote the mutual connection of the continuous part of the HPN model with the discrete part of the HPN model. Namely, the places in the parentheses ($p_{13}$) and ($p_5$) represent, respectively, the alternatives to the places $p_{17}$ and $p_1$. The usage of the particular places depends on the starting situation. Namely, the active $p_1$ means the green color in the first traffic light $L_1$ (in the first direction $D_1$—cf. Fig. 6.3) and the active $p_{17}$ means the red color in the second traffic light $L_2$ (in the crossing direction $D_2$). Likewise, the active $p_5$ means the red color in $L_1$ (in the direction $D_1$) and the active $p_{13}$ means the green color in $L_2$ (in the direction $D_2$).

Having the HPN model, we can simulate the traffic on the intersection completely. We will do this by means of the universal tool HYPENS in Matlab. Namely, HYPENS [26, 27] is able to model timed discrete PN, continuous PN and hybrid PN (more precisely, first-order hybrid Petri nets).

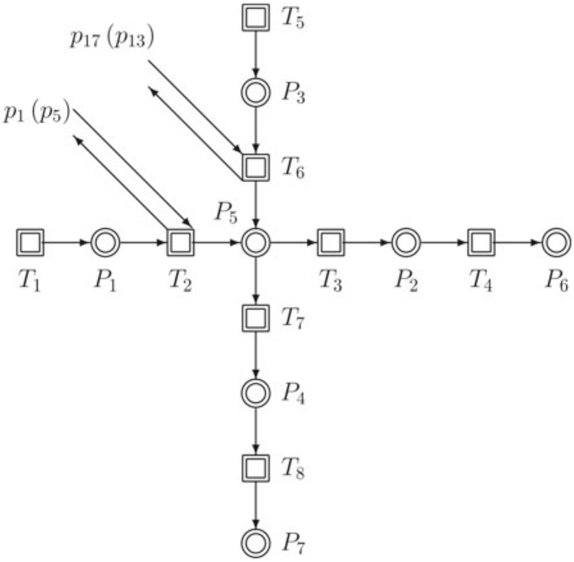


**Fig. 6.6** The HPN model of the intersection depicted in Fig. 6.3

## 6.4 Simulation of Vehicle Flow Propagation Controlled by Traffic Lights

After connecting the TPN model of switching the colors of the traffic lights (corresponding to the P/T PN model displayed in Fig. 6.2) with the model of the roads based on continuous PN given in Fig. 6.6 we are able to simulate the traffic on the crossing roads $D_1$, $D_2$ in time by means of the tool HYPENS in Matlab. In this manner, we can simulate not only the function of the traffic lights $L_1$, $L_2$ but also the corresponding flows of vehicles on the roads $D_1$, $D_2$ controlled by means of the traffic lights. Thus, the cooperation of two modules is concerned. The first module is represented by the TPN model which ensures switching the colors of the traffic lights, while the second module is constituted by the model of the intersection based on continuous PN. The cooperation of these modules in the process of simulation gives us the picture about the vehicle flow propagation with respect to possibilities that are determined by means of the traffic lights.

The simulation results are displayed in Fig. 6.7. Two columns of pictures are introduced there. Each column contains four pictures. They show the flows of vehicles on the roads $D_1$, $D_2$ in front of the traffic lights and behind the traffic lights. Namely, the piecewise-linear courses of flows of vehicles in time for different initial conditions (i.e. for different initial colors of the traffic lights) are displayed there in the graphical form. The left column corresponds to the situation, when in the traffic light $L_1$ the green color shines (in Fig. 6.2 $p_1$ is active), while in the traffic light $L_2$ the red color shines (in Fig. 6.2 $p_{17}$ is active). The right column corresponds to the opposite situation, i.e. when in $L_1$ the red color shines (in Fig. 6.2 $p_5$ is active), while in $L_2$ the green color shines (in Fig. 6.2 $p_{13}$ is active).

When we compare the pictures of both columns each other as well as the pictures inside each column each other we ascertain that the flows of vehicles correspond to switching the colors in the traffic lights. Simultaneously, we ascertain that the flows of vehicles on the roads $D_1$ and $D_2$ are correct. These facts corroborate that the HPN model works correctly.

## 6.5 Generalization

Only the simple segment of the transport system consisting of the simple intersection of two roads was presented above. However, by means of assembling such simple segments more complicated structures of the transport systems can be modelled. The modular approach in building the models is transparent and checking the correctness of the global model is simple. In order to generalize the approach used for the simple intersection introduced in Fig. 6.3 e.g. for the bidirectional flows of vehicles on the roads $D_1$, $D_2$ we can built the HPN model of the bidirectional flows of vehicles given in Fig. 6.8. It is sufficient to use the same module twice. Of course, the second module is used in the opposite direction. However, in such a case a multiplexer

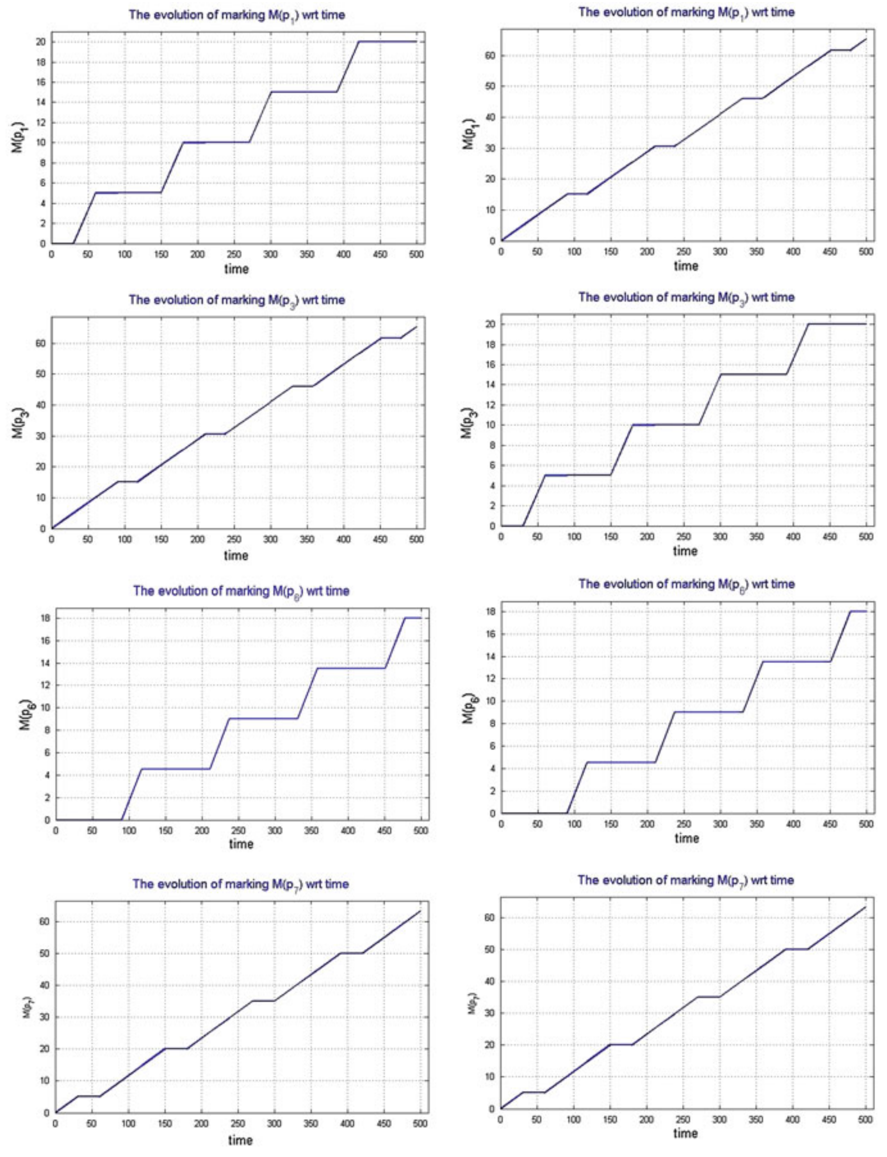


**Fig. 6.7** The simulation results at using the HPN. The evolution of markings of the HPN continuous places $P_1$, $P_3$, $P_6$ and $P_7$ in time representing, respectively, the flows of vehicles in front of the traffic light $L_1$ on the road $D_1$, in front of the traffic light $L_2$ on the road $D_2$, in the road $D_1$ behind the intersection and in the road $D_2$ behind the intersection. The situation when the TPN places $p_5$, $p_{13}$ are connected with the continuous model of the intersection is displayed in the *left column*. The situation when the TPN places $p_1$, $p_{17}$ are connected with the continuous model of the intersection is displayed in the *right column*

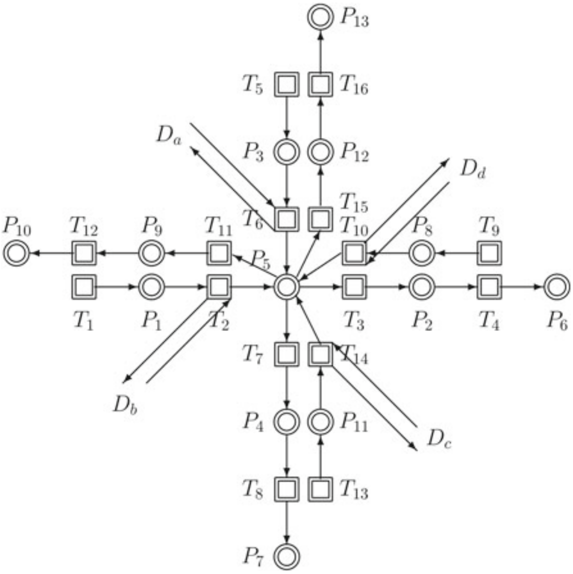


**Fig. 6.8** The HPN model of the intersection of the roads with bidirectional flows of the vehicles

[17] between the traffic lights has to be synthesized. The multiplexer is another important module at the PN-based modelling and control of the transport systems. The general methodology for the synthesis of multiplexers is out of the scope of this chapter. But, a simple TPN model of such a device is sufficient in our case. It can be created e.g. in the form displayed in Fig. 6.9. The symbols $D_a,\ D_b,\ D_c,\ D_d$ indicate

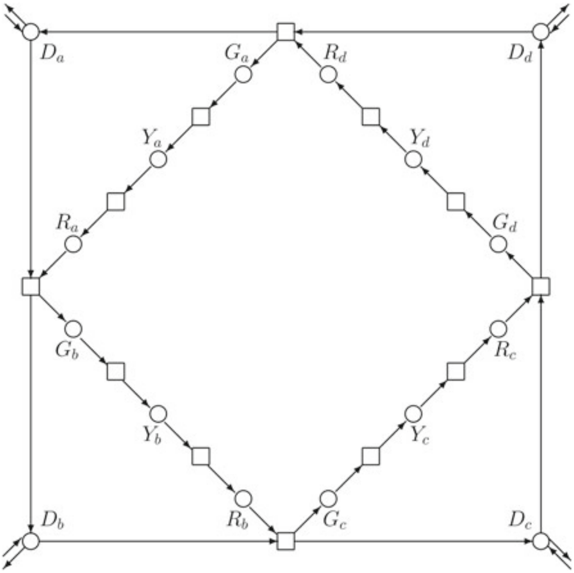


**Fig. 6.9** The structure of the multiplexer of the traffic lights. The symbols $G_x,\ Y_x,\ R_x, x = a,\ b,\ c,\ d,$ express the light colors—i.e. *green*, *yellow* and *red*. $D_a,\ D_b,\ D_c,\ D_d$ denote the places in the corresponding TPN models of the traffic lights

the directions to/from the continuous transitions $T_2$, $T_6$, $T_{10}$, $T_{14}$ placed in Fig. 6.8. They symbolize the places in the TPN model of the multiplexer. Therefore, the same symbols indicate the places in the TPN model of the multiplexer given in Fig. 6.9. Thus, we obtained another structure—the HPN model of the bidirectional traffic on the intersecting roads $D_1$, $D_2$. It is more complicated than the structure given in Fig. 6.3. However, the difficulty of the work with the model during the simulation in HYPENS is the same like in case of the simpler model.

Although any segment is only a relatively simple module, in general more and more complicated structure of the transport system can be built by means of a suitable combination of such modules. In other words, the segments can be interconnected. Such a modular building of the models makes possible to model and simulate complex transport systems. Namely, the interconnected segments may cover e.g. the road structure of a village, a section of a town (ward), the whole town, a district, etc. Consequently, the corresponding PN models can be created. The dynamic cooperation among the adjacent segments (agents) of the transport system will be realized by means of the mutual exchange of the vehicles. The global road structure of the transport system can be modelled and simulated by means of the cooperation of the PN modules corresponding to the individual segments. Videlicet, we can compose bigger PN structures from the PN modules practically arbitrarily.

## 6.6 Conclusion

At present, the vehicle flow propagation in the transport networks represents very important factor of human life. It affects practically all fields of society. Therefore, it is necessary to analyze the matters around the transport systems, and especially around the vehicle flow propagation, carefully. Modelling and simulation help us to do this effectively. Of course, the most important task is to assure the safety of people. The successful control of the vehicle flow propagation by means of traffic lights is an efficient way how to do this. Although there exist different approaches [1–3, 11, 12, 14, 15, 22], the Petri net-based approach was chosen here in order to deal with the modelling and simulation of the vehicle flow propagation in the transport systems. Motivation for this was based especially on the personal experience with different kinds of PN. However, the existence of the (i) methodology for testing the properties of PN; (ii) user friendly tools for modelling and simulation of PN; (iii) possibility to utilize the exact mathematical tool for modelling and control (supervision) of PN; etc. played also a motivating role in deciding what approach will be used. But the most important motivation was the affection to the modularity. Namely, the more complicated PN models can be built from simpler models without any invincible difficulties. The own PN-based approach presented here starts from knowledge acquired from [17, 18, 20, 28–30]. In contrast to these sources just the modularity at building the PN models presented here, in this chapter, is primary. It makes possible to proceed from simpler structures towards more complicated ones.

Three kinds of Petri nets were utilized here in order to model and simulate the segments of the transport system. Namely, P/T PN, TPN and HPN (more precisely FOHPN). Even, it can be said that four kinds of PN were used because HPN consist of two parts—continuous PN and discrete PN (P/T PN and/or TPN). Just the continuous PN can be understood to be the fourth kind of PN used here. The simple transport system segment consisting of the intersection of two roads equipped by the traffic lights given in Fig. 6.3 was studied. At first only the one-way traffic on the roads was taken into account, and afterwards the bidirectional traffic on the roads was examined. Just here the importance of modularity principles was demonstrated, because the cooperation of two simple modules (like that given in Fig. 6.3) creates the more complicated one (given in Fig. 6.8).

Firstly, P/T PN were used for synthesizing the safe and unambiguous structure modelling the traffic lights in the road intersections. The synthesis of the supervisor assuring these properties—see Figs. 6.1, 6.2—was performed by P/T PN too. Also here the modular approach was used. Thus, the unambiguous structure of the P/T PN-based model of two traffic lights was achieved—see Fig. 6.2. The supervisor was synthesized by means of the condition imposed on the Parikh's vector entries. In such a way the priorities between transitions were resolved and the reachability tree without any branching was achieved. Next, the supervised P/T PN model was transformed into the TPN model by means of assigning the time specifications into the P/T PN transitions.

TPN were used here for finding the time relations in the traffic lights placed at the road intersection. The simulation results are shown in Figs. 6.4 and 6.5. While in Fig. 6.4 the deterministic timing of TPN-based model is displayed, the non-deterministic timing (with discrete uniform probability distribution) displayed in Fig. 6.5 shows that in our case no changing the times is necessary. Namely, the courses of the variables in both figures are practically the same.

P/T PN and/or TPN can be utilized in the HPN model (more precisely in the FOHPN model). The continuous part of the HPN model expresses the road network. When this model is used in the simulation process we are able to gain the flows of the vehicles through the intersection. For the HPN model of the simple intersection given in Fig. 6.3 the results introduced in Fig. 6.6 were achieved. The generalization of the HPN model of the intersection towards the more complicated case—the bidirectional traffic in the roads—was shown in Fig. 6.8. The multiplexer of the traffic lights, the TPN model of which is introduced in Fig. 6.9, makes possible to alternate the throughput of the intersecting roads.

Although only a simple isolated segment of the transport system and its doubling (in case of the bidirectional traffic on the roads) were analyzed, modelled and simulated, the achieved results and obtained knowledge may be exploited in more complicated transport network. Namely, at using of the modular approach the individual modules can be interconnected and more complicated structures can be built. Thus, e.g. the road structure of a village, town, district, etc. can be covered by means of the segments and corresponding global PN models can be created. Then, the mutual exchange of the vehicles between the adjacent segments can be enabled.

**Acknowledgments** The theoretical research was partially supported by the Slovak Grant Agency for Science (VEGA) under the current grant # 2/0039/13. The theoretical results were applied at solving the practical project, videlicet: This contribution is the result of the project implementation: Technology research for the management of business processes in heterogeneous distributed systems in real time with the support of multimodal communication, code ITMS: 26240220064, supported by Operational Programme Research & Development funded by the ERDF. The author thanks both institutions for the support of his research.

## References

1. Adler, J.L., Blue, V.J.: A cooperative multi-agent transportation management and route guidance system. Transp. Res. Part C Emerg. Technol. **20**(5–6), 433–454 (2002)
2. Adler, J.L., Satapathy, G., Manikonda, V., Bowles, B., Blue, V.J.: A multi-agent approach to cooperative traffic management and route guidance. Transp. Res. Part B Methodol. **39**(4), 297–318 (2005)
3. Alsop, R.E.: Transport networks and their use: how real can modelling get? Philos. Trans. Royal Soc. A Phys. Math. Eng. Sci. **366**(1872), 1879–1892 (2008)
4. Čapkovič, F.: Automatic control synthesis for agents and their cooperation in mas. Comput. Inform. **29**(6+), 1045–1071 (2010)
5. Čapkovič, F.: Supervision of agents modelling evacuation at crisis situations. In: Jezic, G., Kusek, M., Nguyen, N., Howlett, R., Lakhmi, C. (eds.) Agent and Multi-Agent Systems: Technologies and Applications, LNAI, vol. 7327, pp. 24–33. Springer, Heidelberg (2012)
6. Čapkovič, F.: Assigning jobs to agents by means of Petri net-based models. In: Proceedings of the 14th IEEE International Symposium on Computational Intelligence and Informatics-CINTI 2013, Budapest, pp. 315–320. IEEE, Piscataway (2013)
7. Čapkovič, F.: Petri net-based approach to modelling ATM and minimising logistic costs in ATM network. Int. J. Comput. Intell. Stud. **2**(3–4), 202217 (2013)
8. Čapkovič, F.: Travel routes flexibility in transport systems. In: Barbucha, D., Le, M.T., Howlett, R., Lakhmi, C. (eds.) Advanced Methods and Technologies for Agent and Multi-Agent Systems, pp. 30–39. IOS Press, Amsterdam (2013)
9. Čapkovič, F.: Agent-based modelling the evacuation of endangered areas. In: Nguyen, N., Attachoo, B., Trawiňski, B., Somboonviwat, K. (eds.) Intelligent Information and Database Systems, LNAI, vol. 8397, pp. 281–290. Springer, Heidelberg (2014)
10. Čapkovič, F.: Modelling travel routes in transport systems by means of timed and hybrid Petri nets. In: Jezic, G., Kusek, M., Lovrek, I., Howlett, R., Lakhmi, C. (eds.) Agent and Multi-Agent Systems: Technologies and Applications, Advances in Intelligent Systems and Computing, vol. 296, pp. 97–106. Springer, Heidelberg (2014)
11. Chen, B., Cheng, H.H.: A review of the applications of agent technology in traffic and transportation systems. IEEE Trans. Intell. Transp. Syst. **11**(2), 485–497 (2010)
12. Chen, R.S., Chen, D.K., Lin, S.Y.: ACTAM: cooperative multi-agent system architecture for urban traffic signal control. IEICE Trans. Inf. Syst. E **88D**(1), 119–126 (2005)
13. David, R., Alla, H.: On hybrid Petri nets. Discret. Event Dyn. Syst. Theory Appl. **11**(1–2), 9–40 (2001)
14. De Oliveira, L.B., Camponogara, E.: Multi-agent model predictive control of signaling split in urban traffic networks. Transp. Res. Part C Emerg. Technol. **18**(4), 120–139 (2010)
15. Ebben, M.J.R.: Logistic control in automated transportation networks. Ph.D. thesis, University of Twente, Twente (2001)
16. Iordache, M.V., Antsaklis, P.J.: Supervisory Control of Concurrent Systems: A Petri Net Structural Approach. Birkhäuser, Boston (2006)
17. Jbira, M.K., Ahmed, M.: Computer simulation: a hybrid model for traffic signal optimisation. J. Inf. Process. Syst. **7**(1), 1–16 (2011)

18. Júlves, J., Boel, R.: A continuous Petri net approach for model predictive control of traffic systems. IEEE Trans. Syst. Man Cybern. Part A Syst. Hum. **40**(4), 686–697 (2010)
19. Król, D.: Propagation phenomenon in complex networks: theory and practice. New Gener. Comput. **32**(3–4), 187–192 (2014)
20. List, G.F., Cetin, M.: Modeling traffic signal control using Petri nets. IEEE Trans. Intell. Transp. Syst. **5**(3), 177–188 (2004)
21. Murata, T.: Petri nets: properties, analysis and applications. Proc. IEEE **77**(4), 541–580 (1989)
22. Negenborn, R.R., De Schutter, B., Hellendoorn, H.: Multi-agent model predictive control for transportation networks: serial versus parallel schemes. Eng. Appl. Artif. Intell. **21**(3), 353–366 (2008)
23. Peterson, J.L.: Petri Nets Theory and the Modelling of Systems. Prentice-Hall Inc., Englewood Cliffs (1981)
24. Popova-Zeugmann, L.: Time Petri nets: theory, tools and applications. Part 1. http://www2.informatik.hu-berlin.de/~popova/1-part-short.pdf. Available via Google; cited 13 April 2014 (2008)
25. Popova-Zeugmann, L.: Time Petri nets: theory, tools and applications. Part 2. http://www2.informatik.hu-berlin.de/~popova/2-part-short.pdf. Available via Google; cited 13 April 2014 (2008)
26. Sessego, F., Giua, A., Seatzu, C.: Hypens: a matlab tool for timed discrete, continuous and hybrid Petri nets. In: van Hee, K., Valk, R. (eds.) Applications and Theory of Petri Nets, LNCS, vol. 5062, pp. 419–428. Springer, Heidelberg (2008)
27. Sessego, F., Giua, A., Seatzu, C.: Hypens manual. http://www.diee.unica.it/automatica/hypens/Manual_HYPENS.pdf. Available via Google; cited 12 April 2014 (2008)
28. Soares, M.S., Vrancken, J.: Responsive traffic signals designed with Petri nets. In: Proceedings of the IEEE International Conference on Systems. Man and Cybernetics-SMC 2008, Singapore, pp. 1942–1947. IEEE, Piscataway (2008)
29. Soares, M.S., Vrancken, J.: A modular Petri net to modelling and scenario analysis of a network of road traffic signals. Control Eng. Pract. **20**(11), 1183–1194 (2012)
30. Yaqub, O., Li, L.: Modeling and analysis of connected traffic intersections based on modified binary Petri nets. doi:10.1155/192516. Available via Google; cited 10 March 2014 (2013)

# Chapter 7
# Bio-inspired Routing Strategies for Wireless Sensor Networks

**Pavel Krömer and Petr Musilek**

**Abstract** Successful behavioural and communication strategies of biotic communities can serve as an inspiration for algorithms used to design, manage, and control real-world networks. Many natural systems exhibit complex yet efficient behaviours. Some animal communities display sophisticated behavioural patterns arising from fairly simple activities of their members. The behaviours of ant colonies, swarms of bees, schools of fish, and even some human communities, can be seen as properties of distributed systems consisting of individual agents performing straightforward actions and communicating using simple strategies. Formally, the behaviour of such communities can be modelled as a massive yet intuitive multiagent system. The ensuing models can be applied to a variety of networking problems. This chapter looks at routing in wireless sensor networks and mobile ad-hoc networks as tasks that bear similarities to communication in biotic societies and swarms, and underlines the role of propagation phenomena in routing. It summarizes the basic principles of swarm intelligence and evolutionary computing and reviews recent advances in biologically-inspired network routing.

## 7.1 Introduction

There is a growing need for intelligent protocols and algorithms to design, manage, and control complex cyber-physical systems such as wireless sensor networks (WSN), sensor-actuator networks, mobile ad-hoc networks (MANET) and vehicular ad-hoc networks (VANET). In some applications, such as environmental monitoring [12, 42, 53], these networks are faced with many requirements and challenges

P. Krömer
Faculty of Electrical Engineering and Computer Science, VŠB Technical University of Ostrava, Ostrava, Czech Republic
e-mail: pavel.kromer@vsb.cz

P. Musilek (✉)
Department of Electrical and Computer Engineering, University of Alberta, Edmonton T6G 2V4, Canada
e-mail: petr.musilek@ualberta.ca

D. Król et al. (eds.), *Propagation Phenomena in Real World Networks*,
Intelligent Systems Reference Library 85, DOI 10.1007/978-3-319-15916-4_7

that include autonomous operation, strict energy constraints, low computing power, multi-hop communication, robustness, reliability, adaptability, the ability to operate under harsh environmental conditions, etc. Routing protocols define the strategies and patterns that determine how such distributed networks communicate, and how data propagates [31] from one node to another and eventually outside the network. During the last decade, bio-inspired routing protocols have emerged as a group of methods suitable to address the complex multi-faceted nature of the problem, and specifically to contribute to the energy efficient network routing.

WSN share several properties with MANET and VANET. They are composed of a potentially large number of wireless nodes that perform prescribed tasks and exchange data [12, 17, 35]. Due to their limited transmission range, they communicate in a multi-hop fashion [35, 42]. In MANET, nodes are typically mobile, more powerful and homogeneous, i.e. without distinct roles. WSN nodes, on the other hand, usually have lower computing power, constrained memory, and limited energy available for their operation. Individual WSN nodes have often different roles and utilize diverse hardware (e.g. various types of sensors or energy storage devices). The roles of WSN nodes can be predefined and static, or dynamically assigned in response to the current state of the network or the environment [35, 38, 42]:

- *sensor nodes* collect, store, and eventually communicate data to other nodes. These nodes are the most common, low-power, low-cost devices. They have limited energy available for their operation, consisting in the simple tasks of sampling and transmitting data.
- *relay nodes* play an important role in long distance multi-hop communication. They maintain the connections between WSN segments. Typically, they have more powerful hardware and consume more energy compared to the sensor nodes.
- *sink nodes* are responsible for transmission of data outside the WSN to where it is required. A network can feature one or more sink nodes, sometimes referred to as *base station(s)*.

Individual wireless sensor nodes sometimes have relatively low-precision sensors, but the large number of nodes usually found in sensor networks allows the system as a whole to maintain high spatio-temporal resolution. With the rising complexity of WSN, self-organization and optimization becomes an integral part of their operation [42].

WSN can be used, for example, to monitor indoor or outdoor environments [12, 42]. Typical applications inside buildings include monitoring of temperature, light, humidity, air quality, and a number of safety-related detection tasks (e.g. of smoke or structural deformations). Outdoor applications include monitoring of habitats, environments, agriculture, disaster warning systems, traffic oversight, pollution and water quality assessment, etc. [7].

Most WSN operate according to two main paradigms [12]: *sample and send*, where sensor nodes collect measurements and send them to a sink, and *in-network processing*, where nodes perform additional tasks, such as data aggregation, event detection, or actuation.

Sensing is usually performed by wireless sensor nodes independently. They periodically activate their sensors (e.g. strain, vibration, temperature, or gas sensing devices) and record the measurements characterizing the environment where are they located. This operation is driven by a particular static or dynamic schedule, sometimes called *sensing rate* [46].

A multi-hop wireless transmission is performed to deliver data from wireless sensor nodes to one or more network sinks. Routing algorithms are essential for determining the way data is propagated from one node to another. They have a major influence on important network properties such as communication overhead, data availability (immediacy), network lifetime, and so on. The general objective of WSN routing is to maximize system performance. Performance of a WSN is, however, a broad and rather vaguely defined concept that comprises many sensor/network aspects such as reliability, measurement accuracy, sensor calibration, and error detection [12]. Other routing metrics include communication latency (time needed for a packet to get from its source to a destination), route length, and network energy efficiency. Energy efficiency can be defined, for example, as the ratio of received data and consumed energy [17].

The growth of WSN that can easily scale up to multiple thousands of nodes and the variety of deployment conditions make efficient WSN routing a complex optimization problem. In many cases, this is further exacerbated by the strict energy constraints.

The main traditional routing approaches are [45]:

- *hierarchical routing* that is based on node clustering and role assignments (e.g. Low Energy Adaptive Clustering Hierarchy (LEACH) [24], or Power-efficient Gathering in Sensor Information Systems (PEGASIS) [34]),
- *QoS aware routing* that focuses on achieving quality of service (QoS) requirements, and
- *location based routing* that employs location information for routing purposes.

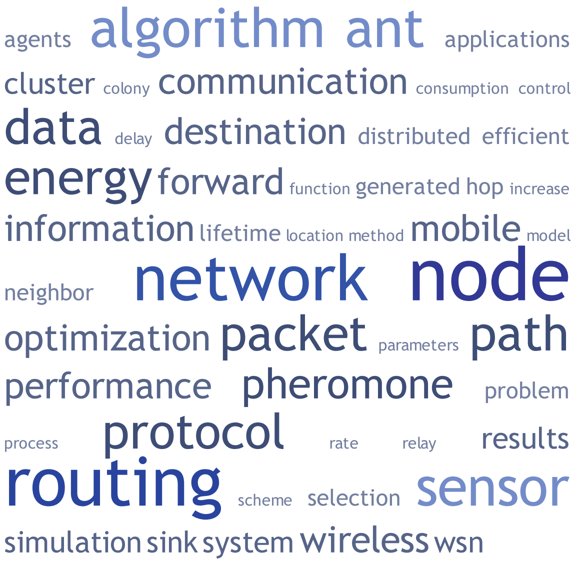
From another perspective, WSN routing algorithms can be classified as [37]:

- *proactive algorithms* that construct data routes independently of the current communication requirements,
- *reactive algorithms* that respond to the communication needs of sensor nodes and construct routes on-demand, and
- *hybrid algorithms* that combine the proactive and reactive methods.

Proactive and reactive routing algorithms suffer from different types of problems. Proactive routing generates large overhead, whereas reactive routing yields high communication delays.

During the last decade, many challenges of MANET and WSN routing have been addressed using various bio-inspired approaches [22, 29, 37, 49, 52]. Bio-inspired methods are often based on a combination of proactive and reactive approaches, allowing them to accomplish adaptive routing, improve network load balancing, and contribute to network topology discovery [37]. As a result, so called *bio-inspired routing* has become a first class WSN routing paradigm [45]. This chapter provides

**Fig. 7.1** Tag cloud formed from the surveyed articles


agents algorithm ant applications
cluster colony communication consumption control
data delay destination distributed efficient
energy forward function generated hop increase
information lifetime location method mobile model
neighbor network node
optimization packet parameters path
performance pheromone problem
process protocol rate relay results
routing scheme selection sensor
simulation sink system wireless wsn

an up-to-date survey of the latest developments in the area of bio-inspired routing in WSN. It complements previous surveys on this and similar topics [37, 42, 49, 61], by considering new application opportunities, new challenges of massive monitoring networks, as well as the emergence of new bio-inspired algorithms. A tag cloud composed of the most frequent, non-trivial terms from surveyed research articles is shown in Fig. 7.1. It clearly illustrates the emphasis on routing, performance, and energy efficiency and the dominance of ant-like algorithms in these papers.

The rest of this chapter is organized into four sections. The bio-inspired methods that are applied in the area of WSN routing most often are reviewed in Sect. 7.2. This includes the description of several methods of swarm intelligence and evolutionary computing, and a short summary of several other bio-inspired methods. Section 7.3 provides an up-to-date, detailed, annotated survey of bio-inspired routing methods. Section 7.4 summarizes the contents of the chapter, brings major conclusions, and outlines future prospects of this exciting research area.

## 7.2 Biologically-Inspired Methods

There are two main groups of bio-inspired methods used to solve WSN routing problems: *swarm intelligence* and *evolutionary computing*. The algorithms based on swarm intelligence [17, 21, 29, 52] include methods inspired by Ant Colony Optimization, Particle Swarm Optimization, and Honey Bee Mating Optimization. Evolutionary methods for WSN routing are mostly based on Genetic Algorithms [2, 17, 37, 49]. The benefits of combining traditional and bio-inspired routing algorithms have been recognized as well [29]. This section provides a brief overview of the principles of swarm intelligence and evolutionary computing, and the description of several selected algorithms.

## *7.2.1 Swarm Intelligence*

Swarm intelligence [9] is a collection of methods to solve complex, real-world problems using the paradigm of collective behaviour of distributed agents. This paradigm has been inspired by the intelligent behaviour of systems composed of many simple individuals, such as ants, bees, bats, etc. Similarly, an artificial swarm system consists of many unsophisticated agents that cooperate in order to achieve desired behaviour [8]. This approach is concerned with exploiting global behavioural patterns emerging from local interactions, rather than with the design of sophisticated central controllers governing the entire system.

### 7.2.1.1 Ant Colony Optimization

Ant Colony Optimization (ACO) [18] is a meta-heuristic approach based on certain behavioural patterns of foraging ants. Ants have shown the ability to find optimal paths between their nests and sources of food. This intelligent path-finding activity is based on stigmergy—indirect communication through modification of the environment. Ants travel randomly to find food, and when returning to their nest, they lay down pheromones. When other foraging ants encounter a pheromone trail, they are likely to follow it. The more ants travel on the same trail, the more intensive the pheromone trace is, and the more attractive it is for other ants.

Emulation of this behaviour can be used as a probabilistic computational technique for solving complex problems that can be cast as finding optimal paths [18]. An artificial ant, $k$, placed on vertex, $i$, moves to node, $j$, with probability

$$p_{ij}^{k} = \frac{\tau_{ij}^{\alpha}\eta_{ij}^{\beta}}{\sum_{l \in N_i^k} (\tau_{il}^{\alpha}\eta_{il}^{\beta})}, \tag{7.1}$$

where $N_{ik}$ represents the neighbourhood of node $i$ for ant $k$ (i.e. a set of nodes that are available for the ant to move to), $\tau_{ij}$ represents the amount of pheromones placed on arc $a_{ij}$, and $\eta_{ij}$ corresponds to a-priori information reflecting the cost of passing the arc $a_{ij}$. After ants finish their forward movement, they return to the nest with food. The tour of ant $k$ is denoted $T^k$. Its length, $C^k$, is used to specify the amount of pheromones, $\Delta\tau_{ij}^{k}$, to be placed by the ant on each arc, $ij$, on the trail that led to the food source

$$\Delta\tau_{ij}^{k} = \begin{cases} \frac{1}{C^k} & \text{if arc } (i, j) \text{ belongs to } T^k \\ 0 & \text{otherwise} \end{cases}, \tag{7.2}$$

$$\tau_{ij} = \tau_{ij} + \sum_{k=1}^{m} \Delta\tau_{ij}^{k}. \tag{7.3}$$

Alternatively, $\Delta\tau_{ij}^{k}$ can be derived from the solution quality expressed as the amount of food collected, $L_k$.

After all ants finish one round of their movement, the amount of pheromones on each arc is reduced through evaporation

$$\tau_{ij} = (1 - \rho)\tau_{ij}. \tag{7.4}$$

The coefficients $\alpha$, $\beta$ and $\rho$ are general parameters of the algorithm that control the ratio between exploitation of known solutions and exploration of new areas of the search space. This canonical form of the ACO algorithm is called Ant System (AS). A pseudocode describing an AS with $n$ ants is shown in Algorithm 7.1.

**Algorithm 7.1** Ant System

```
Generate initial pheromone matrix $\boldsymbol{P}$ with respect to graph topology
$0 \rightarrow generation$
while Termination criteria not met do
    Place $n$ ants randomly on graph vertices.
    Set the amount of collected food for each ant to 0
    foreach ant do
        Move forward on the graph; follow the probabilistic (7.1)
        Compute the amount of collected food corresponding to ants trail
    end
    Find ant with the largest amount of collected food; let the ant lay pheromones in $\boldsymbol{P}$ on its trail
    according to (7.3)
    Evaporate pheromones in $\boldsymbol{P}$ according to (7.4)
    $generation + 1 \rightarrow generation$
end
```

There are numerous variants of the ant algorithm. Modifications of the original ant system, such as elitist ant system and ant colony system [18], max-min ant system, fast ant system, ant-Q, and antabu, have been designed and applied in various problem domains, including bioinformatics, scheduling, data clustering, text mining, and robotics [20]. They have also been successfully used for finding optimal paths in complex networks. They perform best when the problem to be solved has suitable a priori heuristic information, and especially when some sort of local search algorithm is employed [18].

Among bio-inspired routing methods, ACO has been used most often. This can be attributed to their mutual similarity. Multi-hop data transmission is similar to stigmergy in ant communities, a form of communication in massive multiagent systems. Such natural communication strategies have evolved over millions of years and are effective in coordinating colonies of up to several millions of individuals. Emulation of these principles represents a natural choice for the management and control of massive artificial multiagent systems. There are ant-like routing methods for MANET and VANET that focus on improving communication overhead [22], reliability [3], scale [16, 25], and communication cost distribution [28]. Ant-inspired WSN routing methods are primarily motivated by optimization of energy consumption [4, 10, 15, 50]. However, many other objectives have been pursued, including network security [5], self-organization [15], dealing with multiple sinks [38], etc.

#### 7.2.1.2 Particle Swarm Optimization

Particle Swarm Optimization (PSO) is a global, population-based search and optimization algorithm based on simulation of swarming behaviour of bird flocks, fish schools and even human social groups [11, 20, 30]. PSO uses a population of motile candidate particles characterized by their position, $x_i$, and velocity, $v_i$, inside an $n-$dimensional search space they collectively explore. Each particle remembers the best position (in terms of fitness function) it visited, $y_i$, and is aware of the best position discovered so far by the entire swarm $\bar{y}$. In each iteration, the velocity of particle $i$ is updated [20] according to

$$v_i^{t+1} = v_i^t + c_1 r_1^t (y_i - x_i^t) + c_2 r_2^r (\bar{y}^t - x_i^t), \tag{7.5}$$

where $c_1$ and $c_2$ are positive acceleration constants that influence the tradeoff between exploration and exploitation. Vectors $r_1$ and $r_2$ contain random values sampled from a uniform distribution. The position of particle $i$ is updated given its velocity [20] as follows

$$x_i^{t+1} = x_i^t + v_i^{t+1}. \tag{7.6}$$

A basic global PSO (*gbest*) according to [20, 30] is summarized in Algorithm 7.2.

PSO is useful for dealing with problems whose solution can be represented as a point or surface in an $n-$dimensional search space. Candidate solutions (particles) are placed in this space and provided with a random initial velocity. The particles then move through the search space and are periodically evaluated using a fitness function. Over time, particles are accelerated towards those locations in the problem space that have relatively better fitness values.

In addition to the basic model, there is a number of alternative versions of PSO algorithm including self-tuning PSO, niching PSO, and multiple-swarm PSO. These variants have been developed to improve the convergent properties of the algorithm, or to solve other specific problems [11, 20]. A new variant of PSO utilizing the ideas of immune algorithms [40] and orthogonal learning has been recently used to solve

**Algorithm 7.2** *gbest* Particle Swarm Optimization

```
Create population of $M$ particles with random position and velocity
Evaluate an objective function $f$ ranking the particles in the population
while Termination criteria not satisfied do
    for $i \in \{1, \dots, M\}$ do
        Set personal and global best position:
        if $f(x_i) < f(y_i)$ then
            $y_i = x_i$
        end
        if $f(x_i) < f(\bar{y})$ then
            $\bar{y} = x_i$
        end
        Update velocity of particle $i$ by (7.5) and its position by (7.6)
    end
end
```

the challenging task of route recovery and maintenance in networks with mobile sinks [27].

#### 7.2.1.3 Marriage in Honey Bees Optimization

Marriage in Honey Bees Optimization (MBO) [1] is a bio-inspired optimization algorithm that builds on the principles of division of roles and specialization, visual stigmergy, and reproduction strategies found in bee colonies. The principles governing bee colonies are significantly different from those of ant colonies [21]. As the name suggests, the main inspiration for MBO is the complex reproductive behaviour of honey bees.

A honey bee colony consists of a single queen and a number of other individuals that are morphologically uniform, but specialize in different tasks. The single purpose of male bees (drones) is to mate with the queen and contribute to the reproduction of the colony. The drone dies immediately after its role in the mating process has been accomplished [1].

The drones are males born from unfertilized eggs. They are haploid, i.e. they have only one set of chromosomes which is used to fertilize eggs. The mating process involves several mating flights in which the queen mates with several drones and stores their sperm in a special organ called spermatheca [1].

The algorithm considers the mating flight as a set of state-space transitions and probabilistic mating encounters. The queen is initialized with certain energy level and leaves for the mating flight. She returns to the hive when the energy is depleted or her spermatheca full. Drone $d$ mates with queen $q$ with the probability given by an annealing function

$$p_{qd} = e^{-\frac{|f_q - f_d|}{s(t)}}, \tag{7.7}$$

where $p_{qd}$ represents the probability of successful mating (adding the sperm of $d$ to the spermatheca of $q$), $f_q$ and $f_d$ stand for the fitnesses of the queen and drone respectively, and $s(t)$ represents the speed of the queen $q$ at the time of the encounter.

After each transition, the speed, $s(t)$, and energy, $e(t)$, of the queen are decreased using

$$s(t+1) = \alpha s(t), \tag{7.8}$$

$$e(t+1) = e(t) - \Delta_e, \tag{7.9}$$

where $\alpha \in [0, 1]$ is a scaling factor and $\Delta_e$ is a fixed energy reduction step.

When the flight ends, the queen starts breeding by randomly selecting a sperm from the spermatheca and combining it with her genome. The new solution is then subject to random mutation. The algorithm also utilizes a set of worker bees that care of a number of broods [1]. They represent different heuristics that are applied to improve solutions generated by the algorithm. A new solution that has better quality than any of the existing queens replaces the queen in the next iteration. An outline of the generic MBO is shown in Algorithm 7.3.

**Algorithm 7.3** Marriage in Honey Bees Optimization

```
Initialize workers; generate $Q$ queens at random and evaluate them
Use local search to find a good queen
for $flight\ f \in \{1, \ldots, max\_flights\}$ do
    for $queen\ q \in \{1, \ldots, Q\}$ do
        Initialize energy, speed, and position
        while $energy(t) > threshold$ do
            Move $q$ to the next state  Choose drone $d$ according to (7.7)
            if $d$ is selected then
                Add $d$'s sperm to spermatheca
            end
            Update $q$'s energy and speed using (7.8) and (7.9), respectively
        end
    end
    Generate broods by crossover and mutation
    Use workers to improve the broods
    Update fitness and replace low-fit queens by brood with better fitness
end
```

The MBO algorithm combines meta-heuristic and heuristic approaches in a coherent bio-inspired framework. In addition to reproductive behaviour, other aspects of bees' lives have been used as an inspiration for routing algorithms, such as foraging [55], division of labour, and stigmergy [59].

## 7.2.2 *Evolutionary Algorithms*

Evolutionary computing is a group of iterative stochastic search and optimization methods based on the programmatical emulation of successful optimization strategies observed in nature [39]. Evolutionary algorithms use Darwinian evolution and Mendelian inheritance to model the survival of the fittest using the processes of selection and heredity [20].

### 7.2.2.1 Genetic Algorithms

The Genetic Algorithm (GA) is a population-based, meta-heuristic, soft optimization method [39]. GAs can solve complex optimization problems by evolving a population of encoded candidate solutions. The solutions are ranked using a problem specific fitness function. Artificial evolution, implemented by iterative application of genetic and selection operators, leads to the discovery of solutions with above-average fitness. The basic workflow of the standard GA is shown in Algorithm 7.4.

Problem encoding is an important part of the genetic search. It translates candidate solutions from the problem domain (phenotype) to the encoded search space (genotype) of the algorithm. In other words, it defines the internal representation of the problem instances used during the optimization process. The representation specifies the chromosome data structure and the decoding function [13]. The data structure defines the actual size and shape of the search space.

Crossover is the main operator that distinguishes GAs from other population-based stochastic search methods [39]. Its role in GAs has been thoroughly investigated and it has been labeled the primarily creative force in the evolutionary search process. It propagates so called building blocks (solution patterns with above average fitness) from one generation to another, and creates new, better performing, building blocks through their recombination. It can introduce large changes in the population with small disruption of these building blocks [60]. In contrast, mutation is expected to insert new material into the population by random perturbation of chromosome structure. This way, new building blocks can be created or old disrupted [60].

GAs have been successfully used to solve a number of non-trivial multimodal optimization problems. They are capable of effectively searching large, potentially noisy solution spaces. Their clear principles, ease of interpretation, intuitive practical use, and significant results, have made GAs the method of choice for many applications. In the area of WSN routing, GAs have been used towards various objectives including improved energy efficiency [23, 48] and increased network lifetime [6].

**Algorithm 7.4** Genetic Algorithm

```
Define objective (fitness) function and problem encoding
Encode initial population $P$ of possible solutions as fixed length strings
Evaluate chromosomes in initial population using the objective function
while Termination criteria not satisfied do
    Apply selection operator to select parent chromosomes for reproduction: $sel(P_i) \rightarrow parent1$, $sel(P_i) \rightarrow parent2$
    Apply crossover operator on parents with respect to crossover probability to produce new chromosomes: $cross(p_C, parent1, parent2) \rightarrow \{offspring1, offspring2\}$
    Apply mutation operator on offspring chromosomes with respect to mutation probability: $mut(p_M, offspring1) \rightarrow offspring1$, $mut(p_M, offspring2) \rightarrow offspring2$
    Create new population from current population and offspring chromosomes: $migrate(offspring1, offsprig2, P_i) \rightarrow P_{i+1}$
end
```

## *7.2.3 Other Bio-inspired Algorithms*

Swarm intelligence and evolutionary computation are the two major categories of bio-inspired algorithms. However, a number of other biological processes have served as an inspiration for various algorithms. Two interesting bio-inspired methods that have been recently used for WSN routing are based on cell biology and bacterial foraging.

### 7.2.3.1 Cell Biology

Different aspects of cell biology have inspired routing algorithms for WSN, for example consider the recently proposed attractor selection model based on the biology of *E. Coli* [33], and the pheromone protocol inspired by the life of unicellular organism *dictyostelium discoideum* [43].

The attractor selection model [33] provides a bio-inspired mechanism for adaptively selecting one of many possibilities. Each alternative is described by a system of stochastic differential equations modeled after messenger RNA (mRNA) synthesis and degradation. The algorithm converges to solutions that have high value of output probability for one alternative and low output probabilities for all other options.

The second WSN approach rooted in cell biology [43] uses a set of simple local rules describing the behavioural patterns of *dictyostelium discoideum* to implement a simple variant of the well-known swarm intelligence principle of following the path with the best fitness. The unicellular amoeboid *dictyostelium discoideum* produces a chemical signal (a type of pheromone) that attracts other individuals. Wandering *dictyostelia* are sensitive to the pheromone. An individual that detects the signal emits a pheromone itself and moves in the direction with highest pheromone density. This simple indirect communication strategy can be used to solve global optimization problems.

#### 7.2.3.2 Bacterial Foraging Optimization Algorithm

Bacterial Foraging Optimization Algorithm (BFOA) is a recent bio-inspired method [14, 44]. It is a swarm-intelligent algorithm that implements the distinctive food searching strategy of the bacterium *E. Coli* as a type of parallel, nongradient optimization.

*E. Coli* is a simple and common microorganism that has developed a successful survival strategy combining collective and individual decision making. The foraging behaviour involves motility, swarming, reproduction, elimination and dispersal [14, 44]. Motility of the bacteria, called chemotaxis, involves two types of movements: swimming and tumbling. Swimming bacteria move in a fixed direction. Tumbling, on the other hand, is a movement that results in a change of the movement direction. The choice between swimming and tumbling depends on whether the bacteria perceives its environment as favorable. Swarming represents a social self-organization of a group of bacteria that influence each other. Reproduction leads to elimination of the least healthy bacteria and asexual reproduction of the most fit individuals. The best individuals produce clones that initially share their location. Elimination and dispersal serve as simulations of sudden changes in the environment and are implemented as the random removal of existing (healthy) bacteria and the random creation of new ones.

One particular part of the complex BFOA algorithm, the chemotaxis, has been recently used for WSN routing in [26]. It simulates the movement of the bacterium as a series of steps described as follows

$$\theta^i(j+1) = \theta^i(j) + C(i)\frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}}, \tag{7.10}$$

where $\theta^i(j)$ is the position of the $i$-th bacterium at chemotactic step $j$, $C(i)$ is the size of the step taken in the direction of the tumble, and $\Delta(i)$ is a random tumble vector subject to $\forall x \in \Delta(i) : x \in [-1, 1]$. The value of $C(i)$ is selected with respect to the quality of the solution represented by bacterium $i$ located in chemotactic step $j$ at position $\theta^i(j)$. The decision between swimming and tumbling is controlled by a simple logic. If the quality of the solution improves, swim in the current direction; otherwise, tumble. Chemotaxis has been successfully used as the main principle of a recent energy efficient WSN routing algorithm [26].

## 7.3 Bio-inspired WSN Routing Algorithms

This section provides a detailed overview of selected works on bio-inspired routing in the areas of WSN, MANET, and VANET. It is organized in a chronological way to capture the evolution of requirements, algorithms, and results in the field of bio-inspired routing methods.

The use of ACO for network routing can be traced back to 2002. The study by Sim and Sun [56] is an example of an early application of swarm intelligence in the field of general computer networking. It uses ACO to avoid network traffic congestion by continuously updating routing tables. To mitigate stagnation and improve optimization results, the authors use several distinct ant colonies—an approach called Multiple ACO. This study has confirmed that meta-heuristic, agent-based approaches can have different main objectives (e.g. load balancing, QoS). It has also shown that the problem of routing naturally matches the traditional application area of ant-like algorithms originally developed for optimal path finding.

Güneş et al. [22] proposed an ACO-based routing method called Ant Routing Algorithm (ARA) for use in MANETs. The algorithm is designed to achieve robust and reliable on-demand routing in the environment of dynamic wireless networks with mobile nodes. Its goal is to find a multi-hop route between two nodes interested in data exchange. The algorithm operates in two phases. During the *route discovery phase*, special agents called forward ant (FANT) and backward ant (BANT) are used to construct a routing table for the ensuing data exchange. The FANT agents are broadcast to the network by the sender node in a flood-like manner, while the BANT agents are returned by the recipient node. BANTs also mark their route by pheromones in routing tables of the nodes they visit. In the *route maintenance phase*, data packets are routed between the sender and recipient using probabilistic decision rules driven by pheromones in the local routing tables. Continuous updates of the pheromones either maintain or alternate the initial route depending on the actual state of the network. The algorithm operates in a distributed manner relying only on local information. As a result, it enables adaptive on-demand routing with a small overhead. Through software simulations, ARA was compared with traditional MANET routing algorithms such as the Ad-hoc On-Demand Distance Vector (AODV), Destination-Sequenced Distance Vector (DSDV), and Dynamic Source Routing (DSR). The bio-inspired algorithm performed on par with the traditional methods in terms of packet delivery rate and the number of lost packets, but with a lower overhead.

The problem of routing in large scale MANETs was addressed by an ant-like approach in the work of Heissenbüttel and Braun [25]. The authors were interested in an efficient routing in wireless networks covering large geographical areas and comprising of large number of nodes. One of the research objectives was to avoid the flood-like broadcasting of forward agents (as in ARA) in order to scale to large networks. The proposed solution is based on the abstraction of physical network topology and creation of a logical topology employed for routing. Close nodes are grouped together to form *logical routers* (i.e. groups of nodes with identical routing tables), *logical links* between logical routers, and eventually *logical multi-hop paths*. The resulting routing algorithm, called Mobile Ants-based Routing (MABR), uses a single routing table and link cost table for each logical router. Otherwise, its operations are similar to those of ARA [22].

Another ant-based routing algorithm designed specifically for MANET is due to Hussein and Saadawi [28]. The algorithm, named Ant Routing Algorithm for Mobile Ad-hoc Networks (ARAMA), was designed as distributed, self-organizing, and multiobjective. It pays attention to node mobility, elevated error rates, energy constraints, and unbalanced energy distribution in the network. As in the previous cases, one of the main reasons to use a bio-inspired algorithm was the need for a robust procedure that would yield lower overhead than the traditional routing methods and therefore provide better performance and scalability. The basic operations of ARAMA are similar to those of ARA. Forward and backward ants are used to construct on-demand probabilistic routes between sender and receiver. However, the metrics for path evaluation, as well as the intensity of path discovery, are more sophisticated. Path evaluation considers the number of hops, communication delay, quality of service, and node battery state. Path discovery intensity (i.e. the intensity of sending forward and backward ants) is a function of network dynamics. Computer simulations have shown that ARAMA contributes to fair and balanced energy usage across the network.

A routing algorithm called Termite was developed by Roth and Wicker [51] as a robust, bio-inspired procedure for MANET routing. It uses route request and response packets to create and maintain adaptive routing tables. *Route request packets*, assuming the role of forward ants, are propagated through the network in a random walk-like manner until they reach their destination or die. *Route reply packets*, analogous to backward ants, are sent back to the source and update pheromones in routing tables of the nodes they visit. In addition, this algorithm uses two other types of packets: *hello packets* broadcast by nodes when they find themselves isolated, and special *seed packets* that spread the pheromones of each node in an attempt to reduce the need for route construction and to lower the number of route request packets. Stigmergy concepts of linear pheromone updates and exponential pheromone decay are used to keep routing tables up-to-date with changing network topology. However, rather than using a specific bio-inspired method, this algorithm uses a number of general ideas of swarm intelligence.

An adaptive hybrid algorithm designed for routing in networks spread across geographically disperse locations was proposed by Alena and Lee [3] in 2005. The routing strategy utilizes a combination of stigmergy-based probabilistic routing and a-priori information extracted from topographic and radio coverage maps. The ant-inspired portion of the algorithm uses an elitist approach with stronger pheromone updates on best routes found in each iteration. Route discovery and optimization is implemented by periodically sending forward ants to random destinations. Backward ants returning to senders update routing tables of all visited nodes according to the actual state of the network. The algorithm also addresses communication interruptions and performance problems. The goodness of a route is evaluated on the basis of *trip time* which reflects a number of metrics such as number of hops and communication congestion. The algorithm has been evaluated through complex simulations of a network of 50 mobile nodes in a realistic environment described by topographic maps. A comparison of the proposed algorithm with AODV and Ant-AODV has shown that the new algorithm performs best and can be considered for extremely demanding WSN applications such as planetary exploration [3].

Di Caro et al. [16] proposed a complex ant-based routing algorithm for MANETs. The algorithm was based upon another ant-based routing method developed for wired networks. However, it was extended to address the challenges of routing in WSN with dynamic topology and generic motile nodes without predefined roles. In contrast to previous ant-inspired routing methods (e.g. ARA [22]), the proposed *AntHocNet* algorithm aims to achieve a balance between proactive and reactive behaviour to establish a robust stochastic multi-path routing. AntHocNet's reactive *path setup* phase is triggered by data transmission. Forward ants are broadcast to the network with the goal of creating mesh-like (i.e. highly parallel) initial paths. During the *stochastic routing* phase, the packets are sent between the source and destination nodes according to the routing tables, which follow the stochastic path selection rules known from previous ant-inspired routing algorithms. The proactive *path maintenance and exploration* phase of AntHocNet includes periodically sending forward ants to explore new paths and routing configurations. The algorithm also uses several additional optimization techniques to maintain up-to-date routing information: hello messages are used for local communication, and link failure information is broadcast to the network. Computer simulations have shown that the algorithm yields low end-to-end delay and good packet delivery ratio, especially at higher node movement speeds [16]. However, the communication overhead of AntHocNet is higher than that of AODV.

The behaviour of honey bees inspired an energy-aware reactive routing algorithm for MANET introduced by Wedde et al. [59]. The main objective of the algorithm is to maximize network lifetime by distributing communication across multiple paths. This allows to achieve balanced energy consumption without compromising on performance. The algorithm successfully applied honey bee-inspired routing principles originally developed for wired networks in the environment of mobile ad-hoc networks. It is based on a complex and biologically well-described analogy between honey bee foraging strategies and network traffic patterns. In this analogy, every network node is modelled as a virtual *bee hive* with different compartments and different types of bees that travel across the network. Each hive contains bee *packers* who receive packets and transfer them to suitable bee foragers. The *foragers* act as transport agents sensitive to some optimization criteria, such as transport delay or node energy. Foragers also collect information about global network status. Finally, *scout* bees are broadcast to the network in order to discover new routes and share routing information with foragers.

The initial works on bio-inspired routing have shown that inspiration from nature can contribute to the improvement of efficiency and robustness of routing in wireless networks, and particularly in MANETs. They have, however, focused only on some aspects of the routing problem, such as network scaling, congestion elimination, and minimization of routing overhead, but ignored other aspects like energy efficiency and network lifetime.

An energy efficient WSN routing algorithm based on ACO was developed by Camilo et al. [10] in 2006. This study proposed and compared three ant-based routing strategies. *Simple ant routing* is a straightforward proactive routing algorithm utilizing forward and backward ants. The second algorithm, called *improved ant routing*, introduces energy awareness as a part of the route evaluation criteria. The third algorithm, named *energy efficient ant based routing* (EEABR), aims at reducing the amount of information saved in each routing packet (i.e. forward and backward ant). In contrast to previous approaches, each ant contains information about the average energy on the route up to the current node, rather than complete data on energy level for each visited node. The reduction of routing packet size contributes to lower data overhead and higher energy efficiency (ratio of consumed energy to transmitted data packets).

Rahmani et al. [48] developed an agent-based WSN routing strategy using a parallel GA. This approach combines stochastic, cost-based next-hop selection (the probability of sending packet to a neighbour is proportional to the cost of routing to the neighbour) with greedy selection of the neighbour with the highest remaining energy. This allows it to achieve a globally energy-efficient behaviour. A parallel GA is used to find the optimal parameters of the routing function. The search for optimal parameters for certain group of nodes is performed periodically by the base station.

A robust bio-inspired MANET routing algorithm focusing on QoS is due to Leibnitz et al. [33]. Unlike previous approaches, this method is inspired by the microscopic world of cell biology. In particular, it mimics the attractor selection process to adaptively select next-hop nodes. Routing operations of the *adaptive response by attractor selection* (ARAS) algorithm consist of two phases. The *route setup phase* finds a route with minimal number of broadcasts. The *route maintenance phase* follows this route through a probabilistic selection of next-hops, and piggybacks the information about route quality in a process similar to the forward and backward ant transmission in other ant-inspired methods. Software simulations have shown that ARAS has performance comparable to AODV, but with a lower overhead. However, this method has not been compared to other bio-inspired routing methods.

In 2009, Okdem and Karaboga [41] presented an algorithm and hardware platform for ACO-like WSN routing. The use of specialized hardware was motivated by the constrained energy and low processing power of wireless sensor nodes. The main routing objective is the maximization of network lifetime by spreading energy consumption across nodes through multi-path energy-aware routing. In the proposed algorithm, the sender node initializes communication with the base station, while the intermediate nodes are used to relay packets following an energy-aware probabilistic path-selection rule. Up-to-date information about the current status of transmission paths is propagated back to the nodes as a part of acknowledgement packets. Additionally, the nodes propagate their own energy level information to their neighbours. Matlab software simulations have shown that the proposed algorithm is more energy efficient than EEABR [10] when a hardware implementation of the routing chip is used.

A hybrid energy-aware WSN routing algorithm combining self-organized clustering and an improved ACO was presented in [47]. The algorithm uses self-organization to form clusters of nodes, select cluster heads, and establish a cluster head chain for communication between the base station and clusters. The routing algorithm operates in rounds. In each round, new clusters are formed on the basis of node location and energy level. Next, node chains are created by energy-aware ACO within each cluster. Finally, cluster heads representing the clusters are selected. All energy demanding operations are initialized and/or performed by the base station. The routing itself, however, is static and deterministic, following the configuration created by the process outlined above. Software simulations have shown that the proposed algorithm is more efficient than the traditional hierarchical routing algorithm (LEACH [24]).

Bari et al. [6] developed a WSN data gathering schedule that maximizes network lifetime; where lifetime is defined as the time when the first of fixed relay nodes stops operating due to depletion of its energy source. The scheduling task is cast as a global combinatorial optimization problem and solved using GA. To account for energy constraints, the algorithm considers energy dissipation of the sensor nodes. In addition, it can cope with energy depletion or the failure of critical nodes through dynamic rerouting. Compared to traditional techniques, such as integer linear programming, the GA-based scheduling approach can efficiently deal with very large networks.

The performance of two existing ant-based routing protocols under different application scenarios was studied by Domínguez-Medina and Cruz-Cortés [17]. They compared an *ACO-based location-aware routing* (ACLR) utilizing closeness of node neighbours, and the *energy efficient ant-based routing* EEABR [10]. This study simulated different WSN hardware and covered different network scenarios, including balanced versus imbalanced energy allocation, and fixed versus randomly selected source and destination nodes. The study has concluded that EEABR outperformed ACLR in terms of energy consumption, but the use of ACLR resulted in lower latency.

In 2010, Matsumoto et al. [38] proposed a bio-inspired data routing scheme designed to increase WSN lifetime by efficient data gathering, communication balancing, and quick adaptation to changes of network topology. The method was designed with special focus on networks with multiple sinks and large numbers of nodes. Each sink is assigned a distinct pheromone that is propagated to the network via a process called pheromone dispersion. Data transmissions are subsequently routed towards sinks following their pheromones. Software simulations have shown that the proposed algorithm outperforms several other ant-based routing algorithms, as well as a traditional multiple sink-aware routing protocol.

Another bio-inspired routing algorithm for WSN with multiple sinks was proposed by Paone et al. [43]. This protocol, inspired by the behaviour of unicellular organisms, operates in a highly distributed manner without global information about the network. The route construction phase of the protocol is called *signaling*. During signaling, each node spreads its *forwarding attitude* (i.e. information about its ability and willingness to route information towards sinks), and routing tables are created. In the *routing* phase, data packets are routed towards sinks following a probabilistic path selection principle. Computer simulations have shown that the proposed protocol performs better than directed fusion and yields good self-repairing capabilities.

A bio-inspired protocol for balanced packet routing, called BiO4SeL, was proposed in [50] and later extended in [15]. The main objective of the algorithm is to increase network lifetime by distributing data transmission paths between network nodes and base stations with respect to their remaining energy levels. The protocol operates in 3 stages. In the *bootstrap phase*, each node broadcasts information about its energy level. During the *initial route discovery phase*, the base station broadcasts special *iant* packets that discover optimal paths and construct routing tables. By design, BiO4SeL's iant packets avoid construction of long and invalid paths. In the probabilistic *data forwarding* phase, routing tables are used for data transmission and maintained by these packets. Simulations have shown that the method scales well, maintains good packet delivery rate, and achieves the best network lifetime when compared to other relevant routing algorithms such as AODV and ARAMA [28]. It also features low overhead, especially in scenarios with a few nodes producing data.

A routing algorithm for underwater WSN applications was presented by Vieira et al. [57]. The main goal of the protocol is to secure a reliable adaptive route towards mobile sinks in a swarm of mobile sensors monitoring local underwater events in space and time. The underwater environment is especially challenging due to the presence of water currents and because of the properties of underwater acoustic communication. Conventional routing protocols are not suitable for such conditions. One of the main aspects of the proposed protocol is the distribution of location information from sinks (special underwater vehicles or surface buoys) that are equipped by GPS and do not suffer from energy constraints. With the help of location information, the trajectory of mobile sinks on the 2D upper hull of the underwater swarm creates a pheromone trail capturing the position and direction of the underwater nodes. The routing has two principal stages: first, data packets are routed vertically towards the upper hull and then in a 3D cylinder to the location of mobile sinks using a geographic routing protocol.

Villalba et al. [58] proposed an extension to AntHocNet [16] to achieve autonomous operation and self-organization. Like AntHocNet, the new algorithm combines proactive and reactive aspects. However, it prefers routes that share neither nodes nor links (disjoint link/disjoint-node routes). This behaviour is implemented using two distinct types of pheromones called *real pheromones* and *virtual pheromones*. Simulation experiments concentrating on the number of hops in routes have shown that the modified bio-inspired algorithm performs better than AntHocNet in terms of data throughput, transmission success rate, average end-to-end delay, and communication overhead.

Ekbatanifard et al. [19] developed an energy-efficient QoS-routing algorithm using multiobjective GAs. The GA finds the least-cost energy-efficient path that satisfies delay constraints. The objectives optimized by the algorithm include communication reliability (expected number of transmissions for successful data forwarding), energy consumption (sum of energy costs of the routing tree), and the probability that end-to-end delay constraint is met. The algorithm utilizes a traditional multiobjective GA called NSGA2 to find a set of pareto-optimal routing trees.

In 2010, Luo [36] proposed another QoS-based routing algorithm using a GA. It uses a probabilistic q-bit based representation of a set of routes connecting nodes to sinks, mutation and rotation operators, and a fitness function sensitive to communication bandwidth and energy costs. Simulations performed by the author have showed that the proposed approach achieved better throughput, lower delay, and longer network lifetime than hierarchical routing when the number of network nodes was large.

Apparently, energy efficiency became a major topic for the second generation of bio-inspired routing algorithms. Two algorithms, EEABR and AntHocNet, became the de-facto baseline methods for the development of further bio-inspired routing methods. Advanced applications such as routing in networks with multiple sinks, and underwater routing, have also been addressed by bio-inspired algorithms. Advanced variants of bio-inspired algorithms, including parallel and multiobjective GAs and hybrid algorithms, have been used.

A security motivated bio-inspired routing algorithm based on ACO and principles of fuzzy logic was created by Sethi and Udgata [54]. The primary objective of this algorithm is to improve network security and to increase attack resilience. Each node is assigned a *trusted value* determined by a fuzzy inference system from packet drop rate and the ratio between route reply time and time-to-live. This value also becomes part of a probabilistic next-hop selection rule used by an ACO-inspired routing procedure.

Zungeru et al. [61, 62] developed an ACO-based routing protocol for visual sensor networks. It addresses the specific requirements of video and image streaming applications (e.g. high bandwidth). The improved algorithm extending the original EEABR was developed to address the challenges of video surveillance, traffic monitoring, and environmental applications. It improves energy efficiency by intelligent initialization of routing tables, smart route maintenance strategy, prioritization of neighbouring nodes, and reduced frequency of network flooding. The simulations have shown that the new algorithm achieved in visual sensor network applications 30 % lower energy consumption than EEABR. It's performance in other metrics was comparable to that of EEABR.

A hybrid ant-inspired WSN routing algorithm combining the elements of proactive and on-demand routing strategies was presented by Almshreqi et al. [4]. It uses information about average and minimum route energy to find transmission path patterns and to spread out the energy cost of the communication. A simulation has showed that the proposed algorithm performs better than EEABR in terms of total energy consumption and energy efficiency.

Da Silva et al. [55] introduced a new bee-inspired algorithm for data dissemination and propagation. The algorithm extends the traditional hierarchical cluster-based protocols LEACH and LEACH-C with bee-inspired concepts. It is designed for WSN with continuous data flow. Such networks generate and process data continuously rather than in an event-driven manner. Using a realistic energy consumption model, the proposed method applies bee strategies to form clusters and select cluster heads. The algorithm preferably selects nodes located near data sources that have high levels of remaining energy and require very little energy to communicate with the base station. Simulation experiments have shown that this bio-inspired algorithm performs better than both LEACH and LEACH-C in terms of the number of packets sent to the base station, network lifetime, and the time until the first node in the network dies (i.e. the full network coverage time).

In 2012, Hoa and Kim [26] proposed a WSN routing algorithm based on the behaviour of the bacteria *E. Coli*. The algorithm constructs an in-network gradient field with maximum gradient concentration in and around the sink. The gradient of each node is based on its location relative to the sink. When the gradient field is formed, packets are routed using biased random walk. Directional bias of each node is based on its past efficiency as a relay for communication with sink. During the transmission, packets observe the gradient field and *tumble* in case they detect a decrease of the gradient on their route. The advantages of the algorithm include locality (no global information is needed), implicit load-balancing, and energy efficiency. It has been shown that the algorithm performs better than the shortest-path strategy, and yields relatively less mean communication delay and average energy consumption.

A secure ACO-inspired WSN routing algorithm was developed by Alrajeh et al. [5]. The use of a bio-inspired approach was motivated mainly by the need to strengthen the security of the network against network-layer attacks and WSN-specific attacks (such as sinkhole attack, false routing, and so on). The proposed reactive multi-path routing algorithm associates a special *trust value* with each node. The trust value is later used as a part of a decision rule for next-hop selection to route packets along paths that are both efficient and secure. Simulation experiments have demonstrated that the algorithm is performance-wise worse compared to LEACH, but better in terms of security.

Bitam et al. [7] proposed another security-oriented bio-inspired routing algorithm. The hybrid routing algorithm termed HyBR is designed specially for VANET. It incorporates communication patterns inspired by bee swarms applied to vehicle-to-vehicle and roadside-to-vehicle communication. It also utilizes principles of geographic and topology-based routing and employs a GA to find optimum routes. The algorithm has been evaluated by simulations of a realistic road network modelled after the city of Biskra in Algeria. It has been found superior to AODV and the greedy perimeter stateless routing (GPSR) protocols.

A fishing spider-inspired clustering method was used as a part of hierarchical routing algorithm in the work of K. Lee and H. Lee [32]. This algorithm is based on the analogy between water surface waves sensed by the fishing spider and concentric radio signals detected by wireless sensor nodes. The main aim of the algorithm is to form node clusters and select cluster heads on the basis of node neighbourhood degree and remaining energy levels. Such strategy is local and inexpensive, but it reflects the state of the entire network and changes with the environment in order to improve self-organization. Simulations have shown that the algorithm achieved longer network lifetime, higher remaining energy levels, and better scalability than LEACH.

A new genetic approach to energy efficient WSN routing was proposed by Gupta et al. [23]. This new scheme called Genetic Algorithm-based Routing (GAR) applies a GA to find efficient routes in a 2-level WSN. It conducts periodical search for a set of suitable next-hop nodes for each node in the network. The generic GA has been extended with modified crossover and mutation operators. The performance of this routing algorithm has been compared to traditional hierarchical routing and to the algorithm presented in [6]. The GAR algorithm has been found better in terms of achieved network lifetime.

In 2014, Hu et al. [27] proposed a bio-inspired algorithm for efficient routing in WSN with multiple mobile sinks. The method utilizes a multiple-swarm cooperative PSO to quickly recover routes damaged by the movements of the sink. The study introduced a new variant of PSO based on orthogonal immune learning to embrace the problem. The proposed algorithm, in combination with AODV, achieved lower packet loss, lower latency, and higher energy efficiency than three other AODV-based protocols.

The recent years have confirmed the trends observed in the field of bio-inspired routing algorithms. Energy efficiency has become the top optimization goal, especially for routing in an energy constrained WSN. Moreover, network security has emerged as a new aspect of bio-inspired routing. Traditional bio-inspired algorithms including ant colony optimization, genetic algorithms, and particle swarm optimization, as well as hybrid algorithms and new bio-inspired algorithms, have been used to address the routing problems with success.

A mind map created from a citation graph of all surveyed research articles is shown in Fig. 7.2. The elliptical nodes represent bio-inspired approaches, and each rectangular node indicates an article. Each link represents implementation and/or a direct reference between the article and the bio-inspired approach it connects.

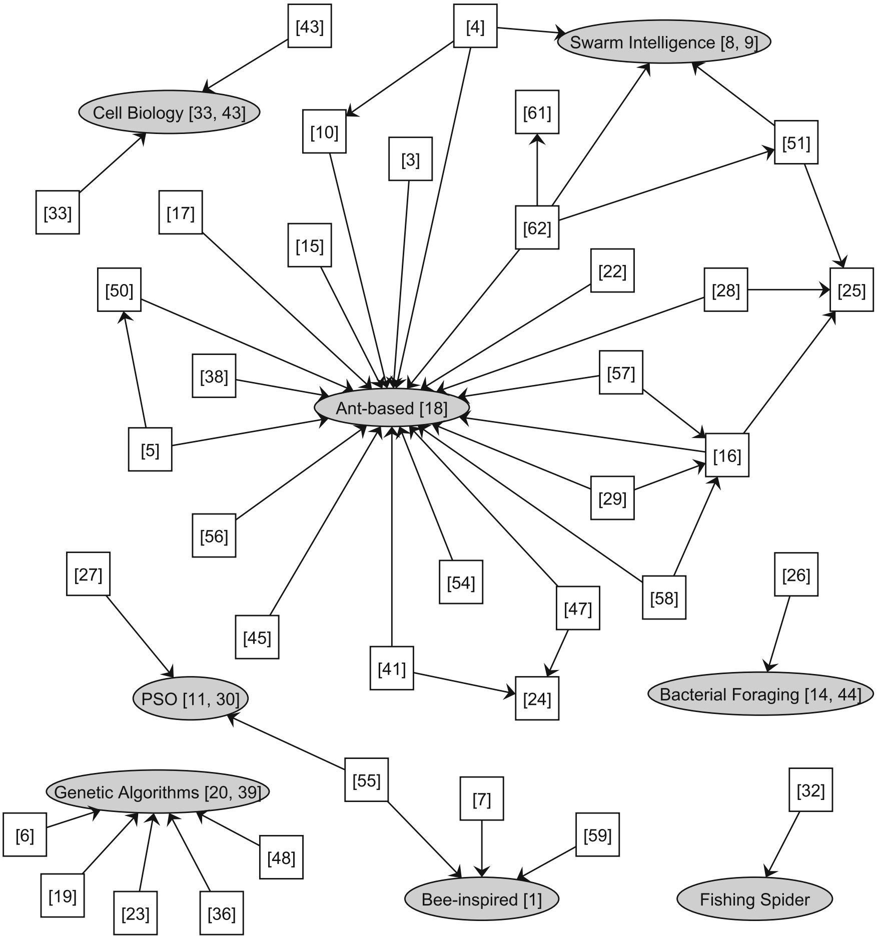


**Fig. 7.2** Mind map illustrating the surveyed bio-inspired routing methods, individual research papers, and their mutual references

## 7.4 Conclusions

This chapter presents an up-to-date survey of recent studies dealing with bio-inspired approaches to routing in wireless sensor networks. Wireless communication is an essential part of WSN operations. Multi-hop data transmission strategies have to be adopted in order to deal with limited communication range of nodes' radios and energy constraints imposed by their limited power sources.

The traditional routing methods are challenged by the growing volume, extensive spatial coverage, and generally high complexity of contemporary WSN. Deterministic, global approaches can hardly address such complex tasks in an efficient way. On the other hand, bio-inspired methods draw their inspiration from natural systems that have properties similar to WSNs. For example, swarm-intelligent algorithms are essentially exchanging information in the way WSN should: in a local and computationally efficient, yet globally highly effective fashion. In contrast, evolutionary approaches can be used for global optimization, even of large scale networks. Various flavours of swarm and evolutionary algorithms, as well as other bio-inspired methods, introduce peculiar strategies that can be applied to achieve diverse routing goals.

In computer applications, bio-inspired methods were first used for optimization and search. As a result, they can be easily used to find communication patterns suitable to satisfy arbitrary objectives (communication latency, throughput, energy efficiency). Bio-inspired methods are also tightly linked to data mining and knowledge discovery. Often, they are used to extract hidden information from data or to uncover implicit patterns. That makes them sensitive to changes in network state and configuration, and thus capable of achieving adaptive routing behaviour.

The limited scope of this chapter could not possibly include all relevant studies. However, it provides a broad and logically structured overview of the most significant trends in bio-inspired routing strategies for wireless sensor networks. The works surveyed in this chapter can be informally classified according to the type of net-

**Table 7.1** Classification of bio-inspired routing algorithms

| Algorithm type | Network type | |
|---|---|---|
| | WSN | MANET/VANET |
| Ant-inspired | [4, 5, 10, 15, 17, 38, 41, 47, 50, 57, 61, 62] | [3, 16, 22, 25, 28, 54, 58] |
| Bee-inspired | [55] | [7, 59] |
| Swarm intelligence | | [51] |
| Genetic algorithm | [6, 19, 23, 36, 48] | |
| Cell biology-inspired | [43] | [33] |
| Bacteria-inspired | [26] | |
| Particle swarm optimization | [27] | |
| Other bio-inspired (fishing spider) | [32] | |

work and the nature of the routing algorithm, as shown in Table 7.1. However, many alternative classifications based on different criteria can be devised:

- the type of the algorithm (from the optimization point of view)—continuous optimization [27] versus discrete optimization (other)
- the type of the routing method—hierarchical routing [32, 47, 55] versus non-hierarchical routing (other)
- the number of sinks in the network—multiple [27, 38, 43, 57] versus single (other)
- the main objective—scalability [15, 16, 25, 32, 50], security [5, 7, 54], energy-efficiency (other)
- etc.

Such classifications are, however, necessarily inaccurate. Some algorithms combine two or more methods and some are suitable for multiple types of networks.

Efficient multi-hop routing is a challenging research problem whose solution is essential for practical operation of the future generation of massive wireless networks. A particular data transmission strategy defines:

- the way *data*, and eventually *information*, is shared among the nodes and propagated across the network, and
- the structure, amount, and propagation patterns of *meta-data* (such as routing tables, logs, or node role assignments), that is used to keep the network operational and its communication efficient.

Meta-data is created and propagated through the network with a single objective: to provide a framework for monitoring the environment and propagating the collected data to the user. From this perspective, the amount of meta-data should be minimized. However, it can be also perceived as a source of complementary information about the network and, indirectly, the monitored environment.

In the bio-inspired analogy, data corresponds to food or fitness, and meta-data is represented, for example, by pheromones, genes or coordinates. Each routing algorithm, mimicking certain biological communication and optimization strategy, affects the organization, scalability, sensitivity, adaptability, speed, and other properties of the network. Lessons learned from nature will undoubtedly be invaluable for attaining the objectives of the next-generation WSN: to facilitate autonomous, accurate, immediate, and cost-effective monitoring of vast heterogeneous environments, with adequate spatial and temporal resolution.

**Acknowledgments** This work has been been supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada, Helmholtz-Alberta Initiative (HAI), and by TECTERRA project 1209-UNI-013

# References

1. Abbass, H.A.: MBO: Marriage in honey bees optimization—a haplometrosis polygynous swarming approach. In: Proceedings of the 2001 Congress on Evolutionary Computation CEC2001, 207–214. IEEE Press, COEX, World Trade Center, 159 Samseong-dong, Gangnam-gu, Seoul, Korea (2001)
2. Adnan, M.A., Razzaque, M.A., Ahmed, I., Isnin, I.F.: Bio-mimic optimization strategies in wireless sensor networks. A Surv. Sens. **14**(1), 299–345 (2013)
3. Alena, R., Lee, C.: Adaptive bio-inspired wireless network routing for planetary surface exploration. In: Aerospace Conference, 2005 IEEE, pp. 1438–1443 (2005)
4. Almshreqi, A., Ali, B., Rasid, M.F.A., Ismail, A., Varahram, P.: An improved routing mechanism using bio-inspired for energy balancing in wireless sensor networks. In: 2012 International Conference on Information Networking (ICOIN), pp. 150–153 (2012)
5. Alrajeh, N.A., Alabed, M.S., Elwahiby, M.S.: Secure ant-based routing protocol for wireless sensor network. Int. J. Distrib. Sens. Netw. **2013**, 9 (2013)
6. Bari, A., Wazed, S., Jaekel, A., Bandyopadhyay, S.: A genetic algorithm based approach for energy efficient routing in two-tiered sensor networks. Ad Hoc Netw. **7**(4), 665–676 (2009)
7. Bitam, S., Mellouk, A., Zeadally, S.: HyBR: A hybrid bio-inspired bee swarm routing protocol for safety applications in vehicular Ad hoc NETworks (VANETs). J. Syst. Archit. **59** 953–967 (2013)
8. Blum, C., Merkle, D.: Swarm Intelligence: Introduction and Applications. Springer Publishing Company, Incorporated (2008)
9. Bonabeau, E., Dorigo, M., Theraulaz, G.: Swarm Intelligence: From Natural to Artificial Systems. Oxford University Press Inc, New York (1999)
10. Camilo, T., Carreto, C., Silva, J.S., Boavida, F.: An energy-efficient ant-based routing algorithm for wireless sensor networks. In: Dorigo, M., Gambardella, L., Birattari, M., Martinoli, A., Poli, R., Stützle, T. (eds.) Ant Colony Optimization and Swarm Intelligence. Lecture Notes in Computer Science, vol. 4150, pp. 49–59. Springer, Heidelberg (2006)
11. Clerc, M.: Particle Swarm Optimization. Wiley, ISTE (2010)
12. Corke, P., Wark, T., Jurdak, R., Hu, W., Valencia, P., Moore, D.: Environmental wireless sensor networks. Proc. IEEE **98**(11), 1903–1917 (2010)
13. Czarn, A., MacNish, C., Vijayan, K., Turlach, B.A.: Statistical exploratory analysis of genetic algorithms: the influence of gray codes upon the difficulty of a problem. In: Webb, G.I., Yu, X. (eds.) Australian Conference on Artificial Intelligence, Lecture Notes in Computer Science, vol. 3339, pp. 1246–1252. Springer, New York (2004)
14. Das, S., Biswas, A., Dasgupta, S., Abraham, A.: Bacterial foraging optimization algorithm: theoretical foundations, analysis, and applications. In: Abraham, A., Hassanien, A.E., Siarry, P., Engelbrecht, A. (eds.) Foundations of Computational Intelligence Volume 3, Studies in Computational Intelligence, vol. 203, pp. 23–55. Springer, Berlin (2009)
15. De Castro, M.F., Ribeiro, L.B., Oliveira, C.H.S.: An autonomic bio-inspired algorithm for wireless sensor network self-organization and efficient routing. J. Netw. Comput. Appl. **35**(6), 2003–2015 (2012)
16. Di Caro, G., Ducatelle, F., Gambardella, L.M.: AntHocNet: an adaptive nature-inspired algorithm for routing in mobile ad hoc networks. Eur. Trans. Telecommun. **16**(5), 443–455 (2005)
17. Domínguez-Medina, C., Cruz-Cortés, N.: Routing algorithms for wireless sensor networks using ant colony optimization. In: Sidorov, G., Hernández Aguirre, A., Reyes García, C. (eds.) Advances in Soft Computing, Lecture Notes in Computer Science, vol. 6438, pp. 337–348. Springer, Berlin (2010)
18. Dorigo, M., Stützle, T.: Ant Colony Optimization. MIT Press, Cambridge (2004)
19. EkbataniFard, G., Monsefi, R., Akbarzadeh-T, M.R., Yaghmaee, M.H.: A multi-objective genetic algorithm based approach for energy efficient QoS-routing in two-tiered wireless sensor networks. In: 2010 5th IEEE international Symposium on, Wireless Pervasive Computing (ISWPC), pp. 80–85 (2010)

20. Engelbrecht, A.: Computational Intelligence: An Introduction, 2nd edn. Wiley, New York (2007)
21. Farooq, M., Di Caro, G.: Routing protocols for next-generation networks inspired by collective behaviors of insect societies: an overview. In: Blum, C., Merkle, D. (eds.) Swarm Intelligence, Natural Computing Series, pp. 101–160. Springer, Berlin (2008)
22. Gunes, M., Sorges, U., Bouazizi, I.: ARA-the ant-colony based routing algorithm for MANETs. In: International Conference on, Parallel Processing Workshops, 2002 Proceedings, pp. 79–85 2002
23. Gupta, S., Kuila, P., Jana, P.: GAR: an energy efficient GA-based routing for wireless sensor networks. In: Hota, C., Srimani, P. (eds.) Distributed Computing and Internet Technology. Lecture Notes in Computer Science, vol. 7753, pp. 267–277. Springer, Berlin (2013)
24. Heinzelman, W., Chandrakasan, A., Balakrishnan, H.: Energy-efficient communication protocol for wireless microsensor networks. In: Proceedings of the 33rd Annual Hawaii, International Conference on, System Sciences, 2000, p. 10 (2000)
25. Heissenbüttel, M., Braun, T.: Ants-based routing in large scale mobile Ad-Hoc networks. In: In Kommunikation in verteilten Systemen (KiVS03), pp. 91–99 2003
26. Hoa, T.D., Kim, D.S.: Bio-inspired and biased random walk routing in dense and lossy wireless sensor networks. In: International conference on Advanced Technologies for Communications (ATC), pp. 247–250 (2012)
27. Hu, Y., Ding, Y., Hao, K., Ren, L., Han, H.: An immune orthogonal learning particle swarm optimisation algorithm for routing recovery of wireless sensor networks with mobile sink. Int. J. Syst. Sci. **45**(3), 337–350 (2014)
28. Hussein, O., Saadawi, T.: Ant routing algorithm for mobile ad-hoc networks (ARAMA). In: Performance, Computing, and Communications Conference, Conference Proceedings of the 2003 IEEE International, pp. 281–290 (2003)
29. Kambayashi, Y.: A review of routing protocols based on ant-like mobile agents. Algorithms **6**(3), 442–456 (2013)
30. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: IEEE International Conference on, Neural Networks, Proceedings, vol. 4, pp. 1942–1948 (1995)
31. Król, D.: Propagation phenomenon in complex networks: theory and practice. N. Gener. Comput. **32**(3–4), 187–192 (2014)
32. Lee, K.H., Lee, H.S.: Energy efficient sensor configuration by fishing spider inspired mechanism. Appl. Mech. Mater. **284**, 2049–2055 (2013)
33. Leibnitz, K., Wakamiya, N., Murata, M.: A bio-inspired robust routing protocol for mobile ad hoc networks. In: Proceedings of 16th International Conference on, Computer Communications and Networks, 2007. ICCCN 2007, pp. 321–326 (2007)
34. Lindsey, S., Raghavendra, C.: Pegasis: Power-efficient gathering in sensor information systems. In: Aerospace Conference Proceedings, 2002. IEEE, vol. 3, pp. 1125–1130 (2002)
35. Lloyd, E., Xue, G.: Relay node placement in wireless sensor networks. IEEE Trans. Comput. on **56**(1), 134–138 (2007)
36. Luo, W.: A quantum genetic algorithm based QoS routing protocol for wireless sensor networks. In: 2010 IEEE International Conference on, Software Engineering and Service Sciences (ICSESS), pp. 37–40 (2010)
37. Marwaha, S., Indulska, J., Portmann, M.: Biologically inspired ant-based routing in mobile ad hoc networks (MANET): a survey. In: Symposia and Workshops on, Ubiquitous, Autonomic and Trusted Computing, 2009. UIC-ATC '09. pp. 12–15 (2009)
38. Matsumoto, K., Utani, A., Yamamoto, H.: Bio-inspired data transmission scheme to multiple sinks for the long-term operation of wireless sensor networks. Artif. Life Robot. **15**(2), 189–194 (2010)
39. Mitchell, M.: An Introduction to Genetic Algorithms. MIT Press, Cambridge (1996)
40. Musilek, P., Lau, A., Reformat, M., Wyard-Scott, L.: Immune programming. Inf. Sci. **176**(8), 972–1002 (2006)
41. Okdem, S., Karaboga, D.: Routing in wireless sensor networks using ant colony optimization. In: First NASA/ESA Conference on, Adaptive Hardware and Systems, AHS 2006, pp. 401–404 (2006)

42. Oliveira, L.M.L., Rodrigues, J.J.P.C.: Wireless sensor networks: a survey on environmental monitoring. JCM **6**(2), 143–151 (2011)
43. Paone, M., Cucinotta, A., Minnolo, A., Paladina, L., Puliafito, A., Zaia, A.: A bio-inspired distributed routing protocol for wireless sensor networks: performance evaluation. In: 2010 IEEE 30th International Conference on, Distributed Computing Systems Workshops (ICDCSW), pp. 247–255 (2010)
44. Passino, K.: Biomimicry of bacterial foraging for distributed optimization and control. Control Syst. IEEE **22**(3), 52–67 (2002)
45. Pavai, K., Sivagami, A., Sridharan, D.: Study of routing protocols in wireless sensor networks. In: International Conference on, Advances in Computing, Control, Telecommunication Technologies, 2009. ACT '09., pp. 522–525 (2009)
46. Prauzek, M., Musilek, P., Watts, A.G., Michalikova, M.: Powering environmental monitoring systems in arctic regions: a simulation study. Elektron. ir Elektrotechnika (2014). (To appear)
47. Qiu, L., Wang, Y., Zhao, Y., Xu, D., Dan, Q., Zhu, J.: Wireless sensor networks routing protocol based on self-organizing clustering and intelligent ant colony optimization algorithm. In: ICEMI '09. 9th International Conference on, Electronic Measurement Instruments, 2009, vol. 3, pp. 223–228 (2009)
48. Rahmani, E., Fakhraie, S., Kamarei, M.: Finding agent-based energy-efficient routing in sensor networks using parallel genetic algorithm. In: International Conference on, Microelectronics, 2006. ICM '06., pp. 119–122 (2006)
49. Ren, H., Meng, M.H.: Biologically inspired approaches for wireless sensor networks. In: Proceedings of the 2006 IEEE International Conference on, Mechatronics and Automation, pp. 762–768 (2006)
50. Ribeiro, L., de Castro, M.: BiO4SeL: A bio-inspired routing algorithm for sensor network lifetime optimization. In: 2010 IEEE 17th International Conference on, Telecommunications (ICT), pp. 728–734 (2010)
51. Roth, M., Wicker, S.: Termite: ad-hoc networking with stigmergy. In: Global Telecommunications Conference, 2003. GLOBECOM '03. IEEE, vol. 5, pp. 2937–2941 (2003)
52. Saleem, M., Caro, G.A.D., Farooq, M.: Swarm intelligence based routing protocol for wireless sensor networks: Survey and future directions. Inf. Sci. **181**(20), 4597–4624 (2011)
53. Selavo, L., Wood, A., Cao, Q., Sookoor, T., Liu, H., Srinivasan, A., Wu, Y., Kang, W., Stankovic, J., Young, D., Porter, J.: LUSTER: wireless sensor network for environmental research. In: Proceedings of the 5th International Conference on Embedded Networked Sensor Systems, SenSys '07, pp. 103–116. ACM, New York, NY, USA (2007)
54. Sethi, S., Udgata, S.: Fuzzy-Based Trusted Ant Routing (FTAR) Protocol in Mobile Ad Hoc Networks. In: Sombattheera, C., Agarwal, A., Udgata, S., Lavangnananda, K. (eds.) Multi-disciplinary Trends in Artificial Intelligence. Lecture Notes in Computer Science, vol. 7080, pp. 112–123. Springer, Berlin (2011)
55. da Silva Rego, A., Celestino, J., dos Santos, A., Cerqueira, E., Patel, A., Taghavi, M.: BEE-C: A bio-inspired energy efficient cluster-based algorithm for data continuous dissemination in wireless sensor networks. In: IEEE International Conference on, Networks (ICON), 2012 18th, pp. 405–410 (2012)
56. Sim, K.M., Sun, W.H.: Multiple ant-colony optimization for network routing. In: Proceedings of the First International Symposium on Cyber Worlds (CW'02), CW '02, IEEE Computer Society, Washington, DC, USA, pp. 0277–0281 (2002)
57. Vieira, L., Lee, U., Gerla, M.: Phero-trail: a bio-inspired location service for mobile underwater sensor networks. IEEE J. Sel. Areas Commun. **28**(4), 553–563 (2010)
58. Villalba, L., Cañnas, D., Orozco, A.: Bio-inspired routing protocol for mobile ad hoc networks. Commun. IET **4**(18), 2187–2195 (2010)
59. Wedde, H.F., Farooq, M., Pannenbaecker, T., Vogel, B., Mueller, C., Meth, J., Jeruschkat, R.: BeeAdHoc: An energy efficient routing algorithm for mobile ad hoc networks inspired by bee behavior. In: Proceedings of the 2005 Conference on Genetic and Evolutionary Computation, GECCO '05, ACM, New York, NY, USA, pp. 153–160 (2005)

60. Wu, A.S., Lindsay, R.K., Riolo, R.: Empirical observations on the roles of crossover and mutation. In: Bäck, T. (ed.) Proceedings of the Seventh International Conference on Genetic Algorithms, Morgan Kaufmann, San Francisco, CA, pp. 362–369 (1997)
61. Zungeru, A.M., Ang, L.M., Prabaharan, S., Seng, K.P.: Ant based routing protocol for visual sensors. In: Abd Manaf, A., Zeki, A., Zamani, M., Chuprat, S., El-Qawasmeh, E. (eds.) Informatics Engineering and Information Science, Communications in Computer and Information Science, vol. 252, pp. 250–264. Springer, Berlin (2011)
62. Zungeru, A.M., Seng, K.P., Ang, L.M.: Chong Chia, W.: Energy efficiency performance improvements for ant-based routing algorithm in wireless sensor networks. J. Sens. **2013**, 17 (2013)

# Chapter 8
# Analysis of Peer-to-Peer Botnet Attacks and Defenses

**Ping Wang, Lei Wu, Baber Aslam and Cliff C. Zou**

**Abstract** A "botnet" is a network of computers that are compromised and controlled by an attacker (botmaster). Botnets are one of the most serious threats to today's Internet. Most current botnets have centralized command and control (C&C) architecture. However, peer-to-peer (P2P) structured botnets have gradually emerged as a new advanced form of botnets. Due to the distributive nature of P2P networks, P2P botnets are more resilient to defense countermeasures. In this chapter, first we systematically study P2P botnets along multiple dimensions: bot candidate selection, network construction, C&C communication mechanisms/protocols, and mitigation approaches. Then we provide mathematical analysis of two P2P botnet *elimination* approaches—index poisoning defense and Sybil defense, and one P2P botnet monitoring technique—passive monitoring based on infiltrated honeypots or captured bots. Simulation experiments show that our mathematical analysis is accurate.

P. Wang
Symantec Corporation, Lake Mary, Florida 32746, USA
e-mail: jenpwang@gmail.com

L. Wu
Department of Computer Science, North Carolina State University,
Raleigh, North Carolina 27695, USA
e-mail: lwu4@ncsu.edu

B. Aslam
National University of Sciences and Technology, Islamabad, Pakistan
e-mail: baber-mcs@nust.edu.pk

C.C. Zou (✉)
Department of Electrical Engineering and Computer Science,
University of Central Florida, Orlando 32816, USA
e-mail: czou@cs.ucf.edu

D. Król et al. (eds.), *Propagation Phenomena in Real World Networks*,
Intelligent Systems Reference Library 85, DOI 10.1007/978-3-319-15916-4_8

## 8.1 Introduction

### *8.1.1 Botnets*

A "botnet" is a network of compromised computers (bots) running malicious software, usually installed via all kinds of attacking techniques such as Trojan horses, worms and viruses. These zombie computers are remotely controlled by an attacker (botmaster). Botnets with a large number of computers have enormous cumulative bandwidth and computing capability. They are exploited by botmasters for initiating various malicious activities, such as email spam, distributed denial-of-service attacks, password cracking and key logging. Botnets have become one of the most significant threats to the Internet.

Today, centralized botnets are still widely used. In a centralized botnet, bots are connected to several servers (called command and control servers) to obtain commands. This architecture is easy to construct and efficient in distributing a botmaster's commands; however, it has a weak link—the command and control (C&C) servers. Shutting down those servers would cause all bots in a botnet to lose contact with their botmaster. In addition, defenders can easily monitor the botnet by creating a decoy to join a specified C&C channel.

In the last few years, peer-to-peer (P2P) botnets, such as Trojan.Peacomm botnet, Storm botnet and its newly improved version Waledac botnet, have emerged as attackers gradually realize the limitation of traditional centralized botnets. "Peer-to-peer botnets" are defined as botnets that rely on peer-to-peer communication mechanisms to facilitate the command and control by their botmasters. There are different ways for a P2P botnet to utilize P2P communication for its command and control. For example, a P2P botnet could use a P2P network (either an existing P2P network, or a unique P2P network formed by its bot members) to directly disseminate its botmaster's commands to all bot members, or it could use a P2P network to disseminate the IP addresses of C&C servers to bot members (like what Storm botnet [53] does, which utilizes an existing P2P protocol to form a hierarchical multi-tier command and control architecture). Due to the fundamental distributive nature of P2P networks, P2P botnets are robust against removal of bots and C&C servers, and have shown great advantages over traditional centralized botnets. As the next generation of botnets, they are more robust and difficult for security community to defend against.

Researchers have started to pay attention to P2P botnet threat in recent years. Trojan.Peacomm botnet, Stormnet and Waledac botnet have been dissected in details in literature [18, 19, 27, 41, 43, 54]. Andriesse et al. [5] reverse engineered P2P botnet Zeus. However, in order to effectively fight against this new form of botnets, it is not enough to simply enumerate and analyze every individual P2P botnet we have encountered in the wild. Instead, we need to study P2P botnets in a more systematic way. Therefore in this book chapter we try to explore the nature of various kinds of P2P botnets, analyzing their similarities and differences, and discussing their weaknesses and possible defenses.

### *8.1.2 Botnet Countermeasures*

From our understanding, botnet countermeasures can be classified into three categories: detection, monitoring, and mitigation.

*Detection* refers to detecting and identifying a botnet in a network. It includes identifying bot members by various ways, such as signature-based malware detection, network flow monitoring, honeypot infiltration, etc.; it also includes discovering C&C channels, such as locating the Internet Relay Chat (IRC) servers of an IRC-based botnet.

*Monitoring* refers to infiltrating a discovered botnet and monitoring its activities. It can help people better understand a botmaster's motivation, a botnet's behavior and design, etc. There are two types of monitoring: active—actively contact bot members to explore their behaviors, and passive—set up traps and wait for bots to contact, such as dark address space monitoring.

*Mitigation* refers to eliminating a discovered botnet, by either curing all/most bots in a botnet or disabling its botmaster's capability in command and control. Upon botnet detection and monitoring, mitigation is the ultimate goal for botnet defense. Because most botnets are large and contain bots located in areas that are beyond our control, in most cases curing all/most bots in a detected botnet is not feasible. Therefore, botnet mitigation usually means isolating bots by disrupting a botnet's C&C channels. This idea can be easily applied to centralized botnets, because in a centralized botnet, the C&C traffic will go through one or a few clearly-defined central servers. As long as we are able to identify the centralized servers of a botnet and stop botnet-related network activities to/from these servers, we can stop the communication between a botmaster and his/her bots, resulting in disabling the botnet. On the other hand, as a P2P botnet utilizes a P2P network to pass important messages across the entire botnet, it is generally much harder to disrupt the information distribution.

There are many research works focusing on botnet detection and monitoring (discussed in Sect. 8.4), but few research works studying botnet mitigation. For P2P botnets, researchers have presented two mitigation techniques—index poisoning defense and Sybil defense [24, 27]. The original ideas behind these two techniques were first introduced to "sabotage"[1] legitimate P2P networks, and now defenders leverage the same ideas to fight against P2P botnets. Empirical studies on index poisoning defense and Sybil defense have been presented in [24, 27], which have shown that they can successfully disrupt the communication of P2P botnets.

In our preliminary study [65], we presented the systematic study of P2P botnets, and provided the mathematical analysis of index poisoning and Sybil defense, but without much discussion and with no simulation evaluation. In this chapter, we study index poisoning defense and Sybil defense techniques further by providing new simulation study and detailed discussions. In addition, we present our new

[1] Index poisoning was introduced by media companies to prevent illegal distribution of copyrighted content in P2P networks [36], while Sybil attack was to subvert a reputation system in P2P networks [17].

investigation on passive monitoring technique, providing both the analytical study of the capability of a monitoring node in a P2P botnet, and the simulation evaluation. To the best of our knowledge, we are the first to provide mathematical analysis on the performance of index poisoning defense, Sybil defense and passive monitoring, not only for P2P botnets, but also for their corresponding attacks targeting general P2P systems. We also confirm the accuracy of our analysis with simulation experiments. We hope to shed light on P2P botnets, and help researchers and security professionals be well prepared and develop effective defenses against them.

### *8.1.3 Contributions*

The major contributions of this chapter are summarized as follows.

- We systematically study P2P botnets along multiple dimensions: infection vectors, bot candidate selection, bootstrap procedure, network structure, C&C mechanisms and communication protocols.
- We mathematically analyze the performance of two popular P2P botnet mitigation techniques: index poisoning defense and Sybil defense.
- We also carefully study passive monitoring of P2P botnets: based on the mathematical analysis of the capability of a monitoring node in a P2P botnet, we are able to provide a lower bound for the number of bots that an infiltrated node can monitor.
- We develop a Kademlia-based P2P botnet simulator. All the analytical results presented in this chapter have been shown to be accurate by simulation-based experiments using this simulator.
- From attackers' perspective, we propose a novel and realistic technique that might be deployed by them to counterattack the index poisoning defense. This method guarantees that command related indices published in a P2P botnet can be generated by and only by botmasters, not by ordinary bots.
- We obtain one counter-intuitive finding: if the index poisoning defense is valid (when a botnet adopts existing P2P protocols and relies on indices to disseminate commands), P2P botnets are equally easy (or hard) to defend compared to traditional centralized botnets.
- The mathematical analysis presented in this chapter is also suitable for modeling index poisoning attack and Sybil attack in legitimate P2P networks, and hence, contribute to the security research for legitimate P2P systems as well.

### *8.1.4 Chapter Organization*

The remainder of the chapter is organized as follows. In Sect. 8.2, we study the life cycle of P2P botnets, which is composed of three stages. Upon our understanding of P2P botnets, a number of countermeasures are presented in Sect. 8.3, and special attentions are given to two mitigation techniques—index poisoning defense (Sect. 8.3.2) and Sybil defense (Sect. 8.3.3), and one passive monitoring technique (Sect. 8.3.4). We review the related work in Sect. 8.4 and conclude in Sect. 8.5.

## 8.2 A Systematic Study on P2P Botnets

The life cycle of botnets is composed of three stages. Stage one—recruiting members, a botmaster needs to compromise many computers in the Internet, so that he/she can control them remotely. Stage two—forming the botnet, bots need to find a way to connect to each other and form a botnet. Stage three—standing by for instructions, after the botnet is built up, all bots are ready to receive communication from their botmaster for further instructions, such as launching an attack or performing an update. In this section, we will discuss each stage in detail.

### *8.2.1 Stage One: Recruiting Bot Members*

P2P networks are gaining popularity in distributed applications, such as file-sharing, web caching, network storage [9]. In these content-trading P2P networks, without a centralized authority it is not easy to guarantee that the contents exchanged are not malicious. For this reason, these networks become the ideal venue for malicious software to spread. It is straightforward for attackers to target vulnerable hosts in existing P2P networks as bot candidates and build their zombie army. Many P2P malware have been reported, such as Gnuman [1], VBS.Gnutella [1] and SdDrop [4]. They can be used to compromise a host and make it become a bot.

However, in this way, the scale of a botnet will be limited by the size of an existing P2P network, and the network will be the only propagation media. On the contrary, P2P botnets we have witnessed in recent years [19, 27, 57] do not confine themselves to existing P2P networks. They have shown that it is more flexible and practical if bot members are recruited from the entire Internet through all possible spread mediums like emails, instant messages and file exchanging.

### *8.2.2 Stage Two: Forming the Botnet*

Upon infection, the next important thing is to let newly compromised computers join the botnet network and connect to other bots. Otherwise, they are just isolated individual computers without much use for botmasters.

Now for the convenience of further discussion, we first introduce three terms: “parasite”, “leeching” and “bot-only” P2P botnets. Each of them represents a class of P2P botnets. In a parasite P2P botnet, all bots are selected from vulnerable hosts within an existing P2P network. The botnet uses this available P2P network for command and control. A leeching P2P botnet refers to one whose members join an existing P2P network and depend on this P2P network for C&C communication, but the bots could be vulnerable hosts that were either inside or outside of the existing P2P network. For example, the early version of Storm botnet [27] belongs to this class of botnet. A bot-only P2P botnet builds its own P2P network, in which all the members are bots, such as Stormnet [27] and Nugache [57].

If all bots are selected from an existing P2P network, it is not necessary to perform any further action to form the botnet, because bots can find and communicate with each other by simply using current P2P protocol. In other words, for a parasite P2P botnet, up to this point, the botnet construction is done and the botnet is ready to be operated by its botmaster.

However, if a random host on the Internet is compromised, it has to know how to find and join the botnet, which is the case for leeching botnets and bot-only botnets. As we know current P2P file-sharing networks provide the following two general ways for new peers to join a network:

1. An initial peer list is hard-coded in each P2P client. When a new peer is up, it will try to contact peers in that initial list to update its neighboring peer information.
2. There is a shared web cache, such as Gnutella web cache [15], stored at some place on the Internet, and the location of the cache is put in the client code. Thus a new peer can refresh its neighboring peer list by going to the web cache and fetching the latest updates.

This initial procedure of finding and joining a P2P network is usually called "bootstrap" procedure. It can be directly adopted for P2P botnet construction. Either a predetermined list of peers or the locations of predetermined web caches need to be hard-coded in the bot code. Then a newly infected host knows which peer to contact or at least where to find candidates of neighboring peers it will contact later.

For instance, Trojan.Peacomm [19] is a piece of malware to create a P2P botnet which uses the Overnet P2P protocol for controlling the bots. A list of Overnet nodes that are likely to be online is hard-coded into the bot's installation binary. When a victim is compromised and runs a Trojan.Peacomm, it will try to contact peers in this predefined list to bootstrap onto the Overnet network. Another P2P botnet, Stormnet [27], uses a similar bootstrap mechanism: the information about other peers with which a new bot member communicates after the installation phase, is encoded in a configuration file that is also stored on the victim machine by Storm worm binary.

### *8.2.3 Stage Three: Standing by for Instructions*

Once a botnet is built up, all bots in the botnet are standing by for instructions from their botmaster to perform illicit activities or updates. Therefore C&C mechanism is very important and is the major part of a botnet design. It directly determines the communication topology of a botnet, and hence affects the robustness of a botnet against network/computer failures, or security monitoring and mitigation.

The C&C mechanisms can be categorized as either *pull* or *push* mechanism. Pull mechanism, i.e., "command publishing/subscribing", refers to the manner that bots retrieve commands actively from a place where botmasters publish commands. On the contrary, push mechanism, i.e., "command forwarding", means bots passively wait for commands to reach them and then forward received commands to others.

For centralized botnets, pull mechanism is commonly used. Take HTTP-based botnets as an example, a botmaster publishes commands on a web page, and bots periodically visit this web page via HTTP to check for any command updates. In the following, we will discuss how pull and push C&C mechanisms can be applied in P2P botnets.

#### 8.2.3.1 Leveraging Existing P2P Protocols

As we discussed above, both parasite and leeching P2P botnets depend on existing P2P networks. Thus it is natural to leverage the existing P2P protocols used by the host P2P networks for C&C communication. Besides, these protocols have been tested in P2P file-sharing applications for a long time, so they tend to be less error-prone than newly designed ones, and have nice properties to improve performance of P2P systems and mitigate network problems, such as link failure or churn ("churn" refers to network dynamics caused by nodes' joining and leaving activities). The following discussion is based on parasite and leeching P2P botnets, but bot-only botnet can adopt these protocols as well.

In P2P file-sharing systems, file index, which is used by peers to locate the desired content, may be centralized (e.g., Napster), distributed over a fraction of the file-sharing nodes (e.g., Gnutella), or distributed over all or a large fraction of the nodes (e.g., Overnet). A peer can send out query message for the file it is searching for, and the message will be passed around according to the routing algorithm implemented in the system. The search will terminate when query hits are returned or the query message expires.

Botmasters can easily adopt the above procedure to disseminate commands in pull-style. They can insert records associated with some predefined file titles or hash values into the index, but rather than putting the content location information, botnet commands are attached. In order to get commands issued by botmasters, bots periodically initiate queries for those files or hashes, and nodes who preserve the corresponding records will return query hits with commands encoded. In other words, bots subscribe the content (i.e., commands) published by their botmaster.

The early version of Storm botnet [27] is a good example to show how a P2P botnet could leverage an existing P2P network or implement an existing P2P protocol for the pull-style command and control, although it uses the Overnet P2P network to pass the locations of its C&C servers instead of botmaster's commands. In Storm botnet, every day there are 32 keys queried by bots to retrieve important information. These 32 keys are calculated by a built-in algorithm, which takes the current date and a random number from [0–31] as input. Therefore, when issuing a command, the botmaster needs to publish it under 32 different keys. Trojan.Peacomm botnet [19] employs the similar design.

Compared to pull mechanism, implementation of push mechanism on existing P2P protocols is more complicated. There are two major design issues:

- Which peers should a bot forward a command to?
- How to forward commands: using in-band (normal P2P traffic) or out-of-band messages (non-P2P traffic)?

To address the first issue, the simplest way is to let a bot use its current known neighboring peers as targets. But the problem of this approach is that command distribution may be slow or sometimes disrupted, because (1) some bots have a small number of neighbors, or (2) some peers in a bot's neighbor list are not bot members in the case of parasite or leeching P2P botnets. One solution to this problem is that letting bots claim they have predefined popular files available, and forwarding commands to peers appearing in the search results for those files. Thus the chance of commands hitting an actual bot is increased. These predefined popular files behave as the watchwords for the botnet, but could give defenders a clue to identify bots.

For the second issue, whether using in-band or out-of-band message to forward a command depends on what the peers in the target list are. If a bot targets its neighboring peers, in-band message is a good choice. A bot could encode a command in a query message, which can only be interpreted by bots, send it to all its neighbors, and rely on them to continue passing on the command in the botnet. This scheme is easy to implement and hard for defenders to detect, because there is no difference between command forwarding traffic and normal P2P traffic. On the other hand, if the target list is generated in other ways, like using peers in returned search results discussed above, bots have to contact those peers using out-of-band message. Obviously out-of-band traffic are easier to detect, and hence, can disclose the identities of bots who initiate such traffic.

The discussion above mainly focused on unstructured P2P networks, where query messages are flooded to the network. In structured P2P networks (e.g., Overnet), a query message is routed to the nodes whose node IDs are closer to the queried key, which means queries for the same key are always forwarded by the same set of nodes. Therefore, to let more bots receive a command, the command should be associated with different keys, such that it can be sent to different parts of the network.

#### 8.2.3.2 Design a New P2P Communication Protocol

It is convenient to adopt existing P2P protocols for P2P botnet C&C communication, however, the inherited drawbacks of existing P2P protocols may limit botnet design and performance. A botnet can be more flexible if it uses a new protocol designed by its botmaster.

The advanced hybrid P2P botnet [63] and the super botnet [61] are two newly designed P2P botnets, whose C&C communication are not dependent on existing P2P protocols. Both of them implement push and pull C&C mechanisms. In a hybrid P2P botnet, when a bot receives a command, it forwards the command to all the peers in its peer list (push), and those who cannot accept connections from others periodically contact other bots in their peer lists and try to retrieve new commands (pull). A super botnet is composed of a number of small centralized botnets. Commands are pushed from one small botnet to another, and within a small centralized botnet, bots pull the commands from their C&C servers. Furthermore, the hybrid P2P botnet is able to effectively avoid bootstrap procedure (required by most of the existing P2P protocols) by (1) passing a peer list from one bot to a host that is infected by this bot, and (2) exchanging peer lists when two bots communicate.

The drawback of designing a new protocol for P2P botnet communication is that the new protocol has never been tested before. When a botnet using this protocol is deployed, the network may not be as stable and robust as expected due to complex network conditions and defenses.

### *8.2.4 Discussion*

Several features can be extracted to represent a P2P botnet during its life cycle: infection vectors, bot candidates, bootstrap procedure, members in the network, communication protocols and C&C styles. Parasite, leeching and bot-only P2P botnets share common features but differ in others, which is summarized in Table 8.1. It is shown that parasite P2P botnets are less flexible but require no bootstrap procedure. This is an advantage of the parasite botnets over the other two classes. Botnets are most vulnerable during bootstrap stage and the propagation could be stopped if bootstrap information is compromised by defenders. Leeching and bot-only P2P botnets are similar, but the former ones are more stealthy. This is because leeching botnets resides in existing P2P networks, resulting in bot members being mixed with legitimate nodes and hard to be detected.

**Table 8.1** Comparison among three types of P2P botnets

| Features | Parasite | Leeching | Bot-only |
|---|---|---|---|
| Infection vectors | P2P malware | Any kind of malware | Any kind of malware |
| Bot candidates | Vulnerable hosts<br>P2P networks | Vulnerable hosts<br>in the internet | Vulnerable hosts<br>in the internet |
| Bootstrap procedure | None | Required | Optional |
| Members in the network | Legitimate peers and bots | Legitimate peers and bots | Only bots |
| Communication protocols | Existing P2P protocols | Existing P2P protocols | Self-designed or existing |
| | | | P2P protocols |
| C&C styles | Pull or push | Pull or push | Pull or push |

## 8.3 Countermeasures

As discussed in Sect. 8.1.2, we believe P2P botnet defense study should be composed with three areas of research: detection, monitoring, and mitigation. Botnet detection has been widely studied by other researchers such as in [10, 28], and hence, we will not discuss it in this book chapter. Instead, we will exploit and analyze possible solutions for P2P botnet monitoring and mitigation.

In research on P2P file-sharing networks, people have long noticed that most P2P protocols are susceptible to index poisoning attack [36] and Sybil attack [17]. Since existing P2P botnets, such as Trojan.Peacomm and Stormnet, directly utilize existing P2P protocols, security defenders could rely on the same principle to conduct index poisoning defense and Sybil defense. In [16, 19, 27], researchers have pointed out that index poisoning defense and Sybil defense can be used to fight against P2P botnets. However, none of them have presented detailed analysis of the performance of these two mitigation approaches, nor have they discussed in detail how attackers might evade these defenses. In this section, we explain how and why these two mitigation approaches work, how attackers can evade them, and give analytical studies to evaluate their performance. Meanwhile, for P2P botnet monitoring, we study and analyze the effectiveness of using a captured bot or an infiltrated honeypot to monitor the members of a P2P botnet.

Before we introduce our analysis of mitigation and monitoring approaches, we will first provide basic background introduction on the Kademlia P2P protocol, which is the protocol used by the famous Trojan.Peacomm and Stormnet P2P botnets considered in our study.

Notations used in this section are summarized and explained in Table 8.2.

**Table 8.2** Parameters used in analysis

| Symbol | | Meaning |
|---|---|---|
| Kademlia | $m$ | The number of bits used to represent a node ID or a key |
| | $k$ | The maximum number of nodes in a bucket in a routing table |
| | $\Delta b$ | The number of bits improved per step for a lookup |
| | $c$ | The number of bits two binary numbers (node ID or key) share in common in their prefixes |
| Botnet | $N$ | The number of nodes in a P2P network |
| | $N_{bot}$ | The number of bots in a P2P network |
| | $N_{tz}$ | The number of bots in the target zone |
| | $N_{index}$ | The number of nodes poisoned in the target zone |
| | $N_{Sybil}$ | The number of Sybil nodes added to the target zone |
| | $N_{query}$ | The number of bots sending out queries for commands |
| | $l_{tz}$ | The length of a search path in the target zone |
| | $P_{success}$ | The probability of a bot getting a real command |

### 8.3.1 Background on Kademlia P2P Protocol

Kademlia is a distributed hash table (DHT) protocol designed for P2P networks [39]. Since it is the protocol implemented in Overnet, a P2P network used by Trojan.Peacomm and Stormnet, this kind of network is our focus in the following sections. Because of page limit, we cannot provide detailed introduction. For more information about Kademlia and Kad, please refer to [39, 51, 58].

In a Kademlia-based network, each node has a unique node ID, which is represented by an $m$-bit binary number. Every node has a routing table containing $m$ lists; each list corresponds to one specific bit of the node ID. Such a list is usually referred as a $k$-bucket, where $k$ is the maximum number of nodes in each list.

Nodes stored in node $A$'s $i$th $k$-bucket ($i = 0, 1, ..., m - 1$) are the nodes whose node ID must have the first $i$ bits in common with node $A$'s ID, but have a different $(i + 1)$-th bit from node $A$'s ID. Figure 8.1 is an example of a routing table on a node whose ID starts with 1011.

In the distributed hash table preserved in Kademlia-based network, each entry is a ⟨key, value⟩ pair, in which the key is also an $m$-bit binary number, and the value part stores the corresponding file or node information. Each ⟨key, value⟩ pair is stored on nodes whose node IDs are the closest ones to the key in the network, and the distance is computed as the exclusive or (XOR) of the key and the node ID. The distance between two nodes is computed in the same way.

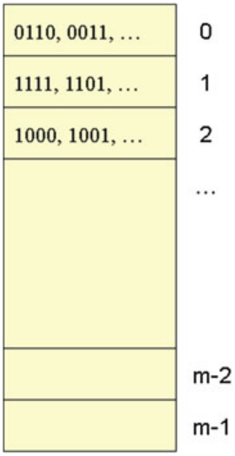


**Fig. 8.1** Routing table of a node whose ID has $m$ bits and starts with 1011 (For illustration purpose, we only use 4-bit prefix to represent each node). The table contains $m$ lists; each list holds at most $k$ nodes and is called “k-bucket”. In the 0th $k$-bucket, the first bit of each node's ID differs from 1011; in the 1st $k$-bucket, each node's ID has the same first bit as 1011, but different second bit. In the 2nd $k$-bucket, nodes share the first two bits with 1011, but have a different third bit. The rest of the $k$-buckets follow the same manner

Kademlia uses iterative routing mechanism. When node $A$ searches for a key, it first finds $\alpha$ nodes that are the closest ones to the key in its routing table, and then initiates lookup queries to these $\alpha$ nodes. Each one of these nodes will send back a response with either the corresponding value part if the $\langle$key, value$\rangle$ pair is stored on it, or a certain number of nodes that are the closest ones to the key in its own routing tables if it does not have the pair node $A$ is looking for. A lookup query stops when there is a query hit or when it expires.

Besides Kademlia, Kad is another popular DHT protocol for P2P networks [2]. It has been deployed by eMule [3] file-sharing application. However, Kad is based on Kademlia with a slightly different routing table structure and parameter settings, such as the number of bits of a node ID (it is 160 in Kademlia, but 128 in Kad). These differences do not affect our study on Kademlia-based P2P network in the following, so our analysis applies to both Kademlia and Kad networks. In the later discussion, we do not differentiate Kademlia from Kad, unless it is explicitly mentioned otherwise.

## *8.3.2 Index Poisoning Defense*

### 8.3.2.1 Defense Idea

Originally, media companies introduced index poisoning attack to prevent illegal distribution of copyrighted content in P2P networks. The main idea is to insert massive number of bogus records into the index system. If a peer receives a bogus record, it could end up not being able to locate the file (nonexistent location), or downloading the wrong file [36].

As we discussed in Sect. 8.2.3, P2P botnets that implement C&C mechanism of command publishing/subscribing make use of the indices in P2P networks to distribute commands. We refer such botnets as “index-based” P2P botnets. If defenders are able to figure out the index keys of the botnet command related index records, they can try to “poison” the index system by announcing false information under the same keys. If the false information overwhelms the real command content, bots that query and retrieve commands will likely end up obtaining false commands. In this way, the C&C channels of the botnet are disrupted.

We believe there are three reasons that index-based P2P botnets are vulnerable to index poisoning defense.

First, a security defect of P2P protocol itself is the root cause. In most P2P networks, there is no central authority to manage the file index system, such that any node, no matter benign or malicious, is able to insert records into the index system. There is no way to authenticate the publishing node and content of the records.

Second, with the help of honeypot and reverse engineering techniques, defenders are able to analyze bot behaviors and bot code, and figure out the bot command related index keys.

Third, in some sense, the C&C architecture of this type of P2P botnets is similar to that of the traditional centralized botnets because of the limited number of index keys

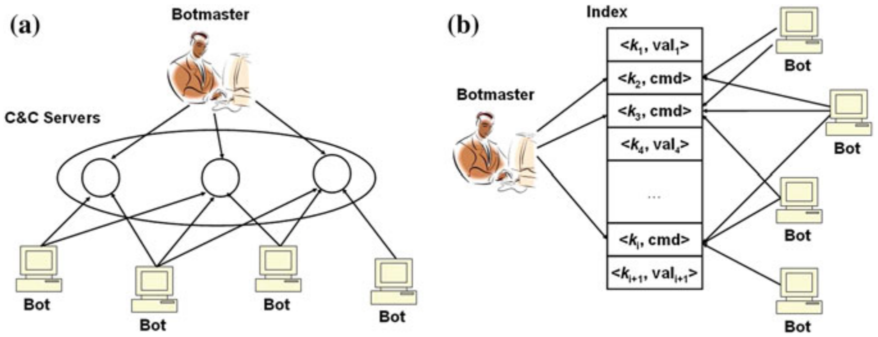


**Fig. 8.2** Similarity of logical C&C structures between traditional centralized botnets and index-based P2P botnets. **a** Centralized botnet. **b** Index-based P2P bonet

for command distribution. As shown in Fig. 8.2, in centralized botnets, commands are published at central sites, where bots are going to fetch the commands; on the other hand, in index-based P2P botnets, commands *cmd* are inserted in the index system by botmasters under special index keys, such as $k_2$, $k_3$ and $k_i$, which are known by bots and queried for retrieving commands.

From the aspect of C&C architecture, index-based P2P botnets logically rely on central points (predefined index keys), while traditional botnets physically rely on central points (predefined C&C servers) for communication. From the aspect of defense, for a traditional C&C botnet, defenders shut down C&C channels by physically removing the C&C servers or blocking access to the servers; while for a P2P botnet, defenders overwhelm real command related records by many bogus records under the same keys (the basic idea of index poisoning technique) to disrupt C&C communication. Therefore, we can draw a conclusion that P2P botnets are not absolutely harder to defend than traditional centralized botnets. If index poisoning defense is valid for a P2P botnet, the P2P botnet is equally easy (or hard) to defend compared with a traditional centralized botnet.

#### 8.3.2.2 Attackers' Possible Counterattack

Although index poisoning defense is effective, it is still possible for attackers to evade it, if they can eliminate the causes discussed in Sect. 8.3.2.1.

Overbot [50], a new botnet protocol designed by Starnberger et al., addressed the second and the third issues. In Overbot, each bot generates its own index key for retrieving command and that key dynamically changes at a certain rate. In addition the communication between bots and sensors (nodes used by a botmaster to publish commands) is encrypted. Thus, it is very difficult for defenders to crack or predict the index key. Even though defenders are able to do it for one single bot, it is not helpful, because different bots have different index keys. However, for the same reason, sensors have to publish a ⟨key, command⟩ pair for each bot they know periodically, which dramatically increases the sensors' workloads and makes them more susceptible to be detected. In other words, the advantages of Overbot come with the cost of introducing scalability and detectability issues.

Now we present a novel and realistic method that attackers might use to deal with index poisoning defense—an authentication enforcement for command generation and index manipulation. It addresses the first cause of index-based P2P botnets being vulnerable to index poisoning (Sect. 8.3.2.1). In this approach, only botmasters can insert records to the command index preserved on bots. Bots can only query to fetch commands.

To realize the authentication, a botmaster generates a pair of public/private keys $\langle K^+, K^- \rangle$, and hard-codes the public key $K^+$ into the bot code. Later, when the botmaster wants to issue a command $m$ under key $k$, he/she can insert a record $\langle k, m, K^-(H(m)) \rangle$ instead of $\langle k, m \rangle$ into the index on a bot, saying bot $A$, where $H(m)$ is the hash value of $m$ (i.e., the command is signed by the botmaster). Bot $A$ can decide if the record is created by its botmaster or not by using the public key $K^+$ to verify the signature. If the signature is authentic, bot $A$ stores this record in the index and waits for others to query, otherwise it discards the fake one. In this way, the index on a bot will not be polluted.

In addition, this authentication mechanism can prevent the spread of false commands. Even if defenders manage to store entries with forged commands in the index on controlled peers (e.g., honeypots infected by a captured bot binary), bots can verify the authenticity of received commands using the public key and disregard the false ones.

Most existing P2P protocols have not implemented this kind of authentication mechanism. Thus in order to deploy it, attackers need to modify the existing P2P protocols, which implies that this counterattack technique can only be applied to bot-only P2P botnets because the botnets cannot join existing P2P file-sharing networks anymore.

#### 8.3.2.3 Analytical Study

In this section, we give an analytical study on performance of index poisoning defense against index-based P2P botnets. Our target is a P2P botnet that implements Kademlia-based DHT protocol for C&C communication. Similar study can be conducted on P2P botnets utilizing other protocols.

As introduced in Sect. 8.3.1, in a Kademlia-based DHT, each entry is a ⟨key, value⟩ pair, and each pair is stored on at least one node whose node ID is closest to the key in the network. If defenders want to pollute a P2P botnet's index records under key $K$, they need to contact nodes (poisoned nodes) whose IDs are close to $K$, and store pairs like ⟨$K$, false value⟩ on them. In this way, when a bot queries for key $K$ to retrieve commands, those poisoned nodes will have a good chance to appear on the search path and return false query value, and hence, prevent bots from reaching nodes who possess the real commands. As illustrated in Fig. 8.3, a bot node $A$ initiates a lookup for index key $K$, the search path is supposed to go through node $B_1$, $B_2$, $B_3$,

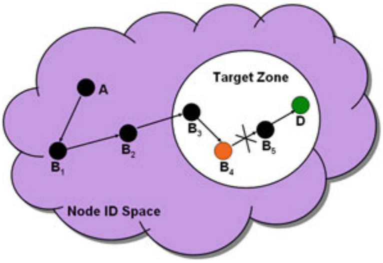


**Fig. 8.3** A search path for a key, where node $A$ is the initiator and node $D$ is the destination. On this path node $B_4$ could be a node targeted by defenders to interrupt the search

$B_4$ and $B_5$, until it reaches node $D$ who has the pair $\langle K$, command$\rangle$. If a pair $\langle K$, false command$\rangle$ has been added in the index on node $B_4$, when the lookup message reaches node $B_4$, the node would return the false command and terminate the search.

We assume that node IDs are uniformly distributed over the entire Kademlia ID space, which is supported by the study in [51]. Suppose defenders choose $N_{index}$ nodes whose IDs have at least the first $c$ bits in common with $K$ to inject and poison their index records. We call the zone around $K$ the "target zone" and all poisoned nodes are in the target zone. Only when a lookup path enters the target zone, it is possible that a poisoned node will be chosen, return a search result and terminate the search. Let $x$ be the probability of choosing a poisoned node in one lookup step, then $1 - x$ is the probability of not choosing one. Therefore the probability of a bot obtaining the real command is

$$P_{success} = (1 - x)^{l_{tz}} \tag{8.1}$$

where $l_{tz}$ is the length of a search path within the target zone.

When a peer initiates a lookup for a key, in general, the expected number of steps required to perform a lookup is given as follows [58]:

$$l = \frac{\log_2 N}{\Delta b} \tag{8.2}$$

where $N$ is the size of the network. We assume all nodes in the P2P network are bots, so $N_{bot} = N$ in this case. $\Delta b$ is the number of bits improved per step, which depends on the structure of the routing table. Thus within the target zone, $l_{tz} = \log_2 N_{tz}/\Delta b$. Since node IDs are uniformly distributed, the number of nodes in the target zone is $N_{tz} = N/2^c$, and $x = N_{index}/N_{tz}$. The complete formula to calculate $P_{success}$ is

$$P_{success} = (1 - \frac{2^c \times N_{index}}{N})^{(\log_2 N - c)/\Delta b} \tag{8.3}$$

According to Eq. (8.3), the performance of index poisoning technique depends on four parameters. We have provided numerical results in Fig. 8.4 to show their impacts on $P_{success}$ by changing the parameters.

Figure 8.4a illustrates that a botnet would be more robust to index poisoning defense, if for each lookup more bits can be improved, i.e., the average length of

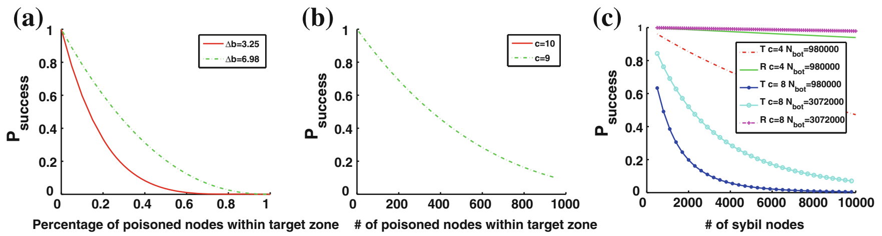


**Fig. 8.4** Performance of index poisoning defense and sybil defense techniques illustrated by numerical results. **a** Index poisoning with different $\Delta b N = N_{bot} = 980{,}000$, $c = 4$. **b** Index poisoning with different $c$ $N = N_{bot} = 980{,}000$, $\Delta b = 3.25$. **c** Sybil attack (T-targeted sybil nodes, R-random sybil nodes) $c = 4$, $\Delta b = 3.25$

search path is shorter. When the search path is short, poisoned nodes have less chance to be chosen along the path.[2]

It is shown in Fig. 8.4b that in order to achieve better performance, defenders could choose a larger $c$, i.e. choosing nodes that are closer to the command related key to poison. However it is not always a good idea to choose a large $c$, because we want to have at least one step along the lookup path in the target zone, otherwise bots can directly get commands without going through any node in the target zone. In other words, $l_{tz} \geq 1$, i.e., $c \leq \log_2 N - \Delta b$. In our case, $N = 980{,}000$, $\Delta b = 3.25$, so $c \leq 16.7$, and for the setup $c = 9$ and $c = 10$ used in Fig. 8.4b, $l_{tz}$ is 3.35 and 3.05 respectively.

The size of the network would also affect the performance of index poisoning defense. However, it does not matter that much, since given a fixed percentage of poisoned nodes in the target zone, it can barely change $l_{tz}$ due to the $\log_2$ operator ($\log_2 980{,}000 = 19.90$ and $\log_2(12{,}000 \times 2^8) = 22.29$).[3]

#### 8.3.2.4 Simulation Evaluation

To evaluate the accuracy of our analysis, we develop a P2P botnet simulator based on OverSim [8], an open source P2P simulator. The P2P botnet we simulate employs Kademlia protocol for the C&C communication.

In [58], Stutzbach et al. defined a system $D(b, r, k)$, which uses $b$-bit symbols with $r$-bit resolution and $k$-buckets, to represent the routing table structure of a Kademlia-based DHT protocol. According to this definition, the routing table structure implemented in our simulator can be denoted as $D(1, 1, 8)$ (i.e., $b = 1$, $r = 1$, $k = 8$), which is consistent with the basic Kademlia design. Correspondingly, the

[2] The value of $\Delta b$ was estimated in [58]. 3.25 is the worst case, while 6.98 is the best case.

[3] An estimate was given in [58] that the Kad network has around 980,000 concurrent peers. Authors of [51] claimed that the population of peers in Kad network is between $12{,}000 \times 2^8$ and $20{,}000 \times 2^8$.

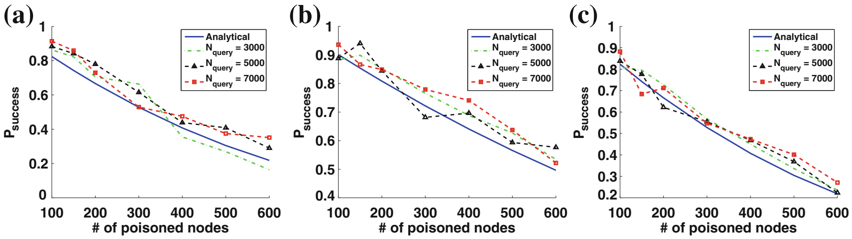


**Fig. 8.5** Comparison between analytical and simulation results of $P_{success}$ for index poisoning defense. The simulated P2P botnet is Kademlia-based with a $D(1, 1, 8)$ routing table structure, and $c = 3$. **a** $N_{bot} = N = 10{,}000$, $m = 16$. **b** $N_{bot} = N = 20{,}000$, $m = 16$. **c** $N_{bot} = N = 10{,}000$, $m = 160$

average bits improved per lookup step in our simulated system is $\Delta b = 4.41$.[4] To reduce the computation time, we set $m$ to be 16 instead of 160 which is the default setting in Kademlia protocol. Experiments on comparing the performance of index poisoning defense with different values of $m$ (Fig. 8.5a, c) show that the value of $m$ does not matter.

We consider two different sizes of such botnets with $N = 10{,}000$, and $N = 20{,}000$, respectively. The parameters we change are $N_{query}$, the number of bots who queries for commands and $N_{index}$, the number of bots whose indices have been poisoned. The node IDs of these $N_{index}$ poisoned nodes share at least the first $c = 3$ bits with a given command related key. In each simulation run, every bot in the set of $N_{query}$ query bots looks for the command once; and we calculate $P_{success}$ based on how many of them actually obtain the real command. To derive the average value of $P_{success}$, we conducted at least 20 simulation runs for each botnet configuration.

Figure 8.5 shows the experiment results comparing to the analytical result obtained from our analysis. According to Eq. (8.3), $P_{success}$ does not depend on $N_{query}$. Therefore, only one curve is plotted as the analytical result (the solid blue line in the figure). As we can see, the analytical result matches with simulation results of $P_{success}$ with around 10 % of errors. Figure 8.5c plots the results from another simulation with the same settings as Fig. 8.5a, except that $m = 160$. According to our analysis, Eq. (8.3), $P_{success}$ does not depend on the value of $m$. Figure 8.5c confirms this conclusion.

## 8.3.3 *Sybil Defense*

### 8.3.3.1 Defense Idea

In a normal P2P file-sharing network, "Sybil attack" is referred as the forging of multiple identities by attackers to subvert the reputation system [17]. The reason of

[4] Please refer to the paper [58] for the detailed formulas to compute $\Delta b$ given the routing table structure $D(b, r, k)$.

P2P networks being vulnerable to Sybil attack is that peers can join the network without authentication or validation of their identities. It is an inherent vulnerability for most P2P networks and protocols [52, 64].

For the same reason, an index-based P2P botnet that implements a traditional P2P protocol will also be susceptible to Sybil-based defense as well. With the knowledge of index keys used for command distribution, defenders can add Sybil nodes (such as honeypots) into the botnet to re-route or monitor the command related traffic. How to set up Sybil nodes depends on the actual P2P system implementation. In an unstructured P2P network, in order to capture more botnet traffic, defenders will set up Sybils to be peers with more important roles, e.g. setting up Sybil nodes as ultrapeers in Gnutella because only ultrapeers are allowed to forward messages. In a structured P2P network, such as Kad, the node IDs of Sybil nodes should not be chosen randomly, but be close to a known command related index key, as discussed in [16, 27]. In this way, command query traffic for the key will go through Sybil nodes with a high probability according to the Kad's routing algorithm. We call such defense "targeted" Sybil defense.

For defenders, the cost for Sybil defense is usually higher than index poisoning defense. This is because either a physical or a virtual machine is needed to set up a Sybil node; in other words, more Sybil nodes require more computer resources, while publishing different records to poison index system can be done by a single node.

#### 8.3.3.2 Attackers' Possible Counterattack

Similarly, approaches used for protecting today's P2P networks from Sybil attack may also work for botmasters to prevent defenders from infiltrating their P2P botnets using Sybil nodes. Here we briefly introduce possible counterattack methods.

In Kademlia-based P2P networks, a node ID can be constructed by hashing the node's IP address as what Chord does [55], rather than being randomly generated by a joining node itself like what Kad does [51]. If the network uses a node's IP address to generate the node ID, Sybil nodes will not be able to choose any IDs they want. When a botmaster applies this scheme in his/her P2P botnet, defenders cannot target a specific key to set up their Sybil nodes. In this case, Sybil nodes are just randomly added into the botnet, which is referred as "random" Sybil defense. This kind of Sybil defense is much less effective than targeted Sybil defense as explained in the next section.

Furthermore, caching technique [39], which was meant to solve "hot spots" problem, can also be utilized by a P2P botnet to reduce the effectiveness of Sybil defense. Because the command related index records will be stored not only on bots that were chosen at the beginning by their botmaster (e.g., unstructured network) or according to the protocol (e.g., structured network), but also on bots that may not be easily identified. Thus even targeted Sybil defense cannot cover all the bots that possess the command information.

#### 8.3.3.3 Analytical Study

Now we analyze Sybil defense on the same type of P2P botnets as in Sect. 8.3.2.3, Kademlia-based P2P botnets. The notations have the same meaning unless explicitly mentioned otherwise.

If node IDs can be chosen randomly, defenders can create special $N_{Sybil}$ Sybil nodes, whose node IDs share at least the first $c$ bits with an index key $K$, and add them into the botnet. Once a Sybil node is on the path of a command lookup, it can re-route the message or return a false command and terminate the search, and hence, prevent the query bot from obtaining the real command.

As we can see, Sybil defense shares the same defense principle with index poisoning defense. They both try to manipulate the command lookup path, as shown in Fig. 8.3. Sybil defense achieves this manipulation by adding new special nodes (controlled by defenders) to the network, i.e., node $B_4$ in Fig. 8.3 is a Sybil node added by defenders, while index poisoning defense achieves this by poisoning the nodes (bots probably) already in the network.

Following the same analysis procedure as what we used in Sect. 8.3.2.3, the probability of a bot successfully getting the real command $P_{success}$ can be calculated using Eq. (8.1), except that $x$ becomes the probability of choosing a Sybil at each step along the search path within the target zone, which is $N_{Sybil}/(N_{Sybil} + N_{tz})$. So

$$P_{success} = (1 - \frac{N_{Sybil}}{N_{Sybil} + N_{tz}})^{l_{tz}} \tag{8.4}$$

where $l_{tz} = \log_2(N_{Sybil} + N_{tz})/\Delta b$, and $N_{tz} = N_{bot}/2^c$.

Differing from what used in the index poisoning defense analysis, the size of the network used in Sybil defense analysis is not the number of the bots, but the total number of bots and Sybil nodes, i.e., $N = N_{bot} + N_{sybil}$, since Sybil nodes added by defenders are not real bots.

When a verification mechanism for node ID is applied in the botnet (Sect. 8.3.3.2) such that defenders can only conduct random Sybil defense, the whole network becomes the target zone, i.e., $N_{tz} = N_{bot}$. Simply substituting $N_{tz}$ in Eq. (8.4) by $N_{bot}$, we can get the following formula to compute $P_{success}$ in this "random" Sybil defense.

$$P_{success} = (1 - \frac{N_{Sybil}}{N_{Sybil} + N_{bot}})^{\log_2(N_{Sybil}+N_{bot})/\Delta b} \tag{8.5}$$

It is shown in Fig. 8.4c that under the same circumstance targeted Sybil defense greatly outperforms the random one. This is because in the former case, Sybil nodes with specially chosen IDs have more chances to appear along a search path than those in the latter case. With limited resources that defenders may use to launch Sybil defense, if the Sybil nodes are closer to the key $K$ (i.e., larger $c$), the defense performance would be better. Furthermore, Sybil defense is more effective if the network is smaller.

#### 8.3.3.4 Simulation Evaluation

We use simulation experiments as well to verify our analysis. The network settings and parameter configurations are the same as those used in Sect. 8.3.2.4.

We assume $N_{Sybil}$ Sybil nodes, whose node IDs share at least the first 3 bits ($c$ =3) with a given key, are added by defenders during the construction of the botnet. When the botnet is built up, the whole network has $N = N_{bot} + N_{Sybil}$ nodes, and $N_{query}$ bots will start querying for commands. Again, we use 20 simulation runs to obtain the average simulation results. The simulation results along with the numerical results are plotted in Fig. 8.6, which shows that our analysis is consistent with the simulations.

### *8.3.4 P2P Botnet Passive Monitoring*

Botnet monitoring is an important component in the overall botnet defense. A good monitoring could collect valuable information about the botnet under observation, such as the size of the botnet, the unique features of the botnet network traffic, and the identities of bots, etc.

Monitoring systems can be classified as either active or passive. Active monitoring usually starts with one or a couple of known bots within the network. By actively contacting these bots, defenders could get to know the identities and information of more bots, and make contacts with those newly discovered bots in the next round. The monitoring is done actively and iteratively until no more unknown bots can be discovered. Passive monitoring is carried out by Sybil nodes put by defenders in the botnet. Unlike active monitoring, these nodes do not actively contact other nodes in the network; they only perform the routine tasks like other normal nodes, such as forwarding traffic and responding to queries. Nodes that have contacted the monitoring nodes are recorded for further analysis. Passive monitoring has the

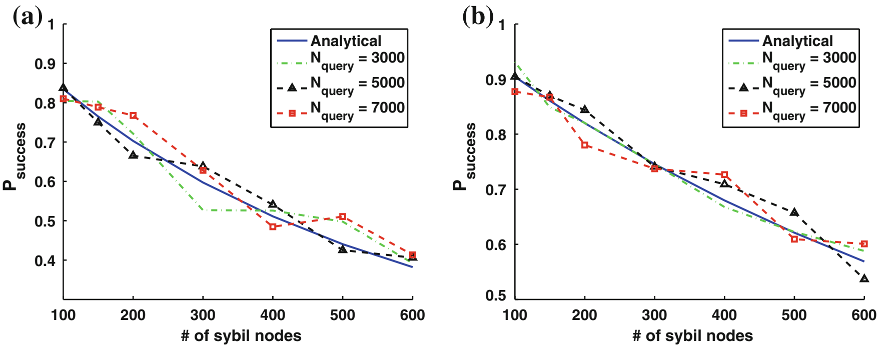


**Fig. 8.6** Comparison between analytical and simulation results of $P_{success}$ for Sybil defense. The simulated P2P botnet is Kademlia-based with a $D(1, 1, 8)$ routing table structure, and $c = 3$. **a** $N_{bot} = 10{,}000$. **b** $N_{bot} = 20{,}000$

advantage of being stealthy, and hence, harder for botmasters to detect and remove those monitoring nodes from their botnets.

In this section, we provide mathematical analysis of the effectiveness of passive monitoring. In other words, we would like to figure out how many bots in a P2P botnet a passive monitoring node can monitor after a certain time period. In this section we address this problem in a Kademlia-based P2P botnet as well.

#### 8.3.4.1 Analytical Study

Suppose defenders have set up one passive monitoring node in a P2P botnet. We want to estimate the number of bots that have this monitoring node in their routing tables, denoted as $N_{routing}$. According to Kademlia protocol, a node would contact nodes in its routing table from time to time because of query or routing table refresh activities. Therefore, $N_{routing}$ is the lower bound for the number of bots that can be observed by a passive monitoring node.

In a Kademlia-based P2P botnet with $N$ nodes, we denote each node as $B_i$, where $i = 1, 2, ..., N$, and the time of node $B_i$ joining the network is denoted as $t_i$, where we assume $t_i < t_j$, if $i < j$, and $t_i \neq t_j$, if $i \neq j$, i.e., no two nodes join the network at the same time. Moreover, when node $B_i$ joins the network, the current size of the network is denoted as $N_i$. Because of the way we index the nodes, we can easily know that $N_i = i$ .

To compute the number of nodes who have a specific node $B_i$ in their routing tables, we need to consider two types of nodes: the nodes joining the network before $B_i$, referred as $Nodes_{before}$, and the nodes joining the network after $B_i$, referred as $Nodes_{after}$.

When node $B_i$ joins the network, there are already $N_{i-1} = i - 1$ nodes in the network. We can classify these $N_{i-1}$ nodes into $m$ groups. The $c$-th group ($c = 0, 1, 2, \cdots, m-1$) contains the nodes whose IDs share the first $c$ bits with node $B_i$'s ID but differ at the $(c+1)$-th bit. Because node IDs are uniformly distributed, the number of nodes in the $c$-th group is $N_{share}(c) = N_{i-1}/2^{c+1}$. If a node in the $c$-th group whose $c$-th bucket is not full (i.e., $N_{share}(c) \leq k$), it will add node $B_i$ in this bucket, otherwise it will not contain node $B_i$ in its routing table. As $c$ increases, the size of $c$-th group monotonously decreases. When $0 \leq c < c_0$ where $c_0 = \lceil \log^{N_{i-1}/k} - 1 \rceil$ ($c_0$ is obtained by letting $N_{share}(c) = k$), $N_{share}(c) > k$, and hence, we do not need to consider nodes in these groups. Therefore the number of $Nodes_{before}$ who would add node $B_i$ into their routing tables can be calculated as follows:

$$N_{before}(i) = \sum_{c=c_0}^{m-1} \frac{N_{i-1}}{2^{c+1}}, \tag{8.6}$$

where $c_0 = \lceil \log^{N_{i-1}/k} - 1 \rceil$, which is obtained by letting $N_{share} = k$.

After node $B_i$ has joined the network, for the nodes joining in later on, they may add node $B_i$ into their routing tables as well. Let's consider a node $B_j$, $i < j$, the probability of these two nodes' IDs sharing the first $c$ bits but differing at the $(c+1)$-th bit is

$$P_{share}(c) = \frac{2^{m-(c+1)} - \frac{N_j}{2^{c+1}}}{2^m - N_j} = \frac{1}{2^{c+1}}, c = 0, 1, ..., m-1. \tag{8.7}$$

Suppose node $B_i$ and node $B_j$ share the first $c$ bits but differ at the $(c+1)$-th bit in their IDs, there are $N_{share}(c) = N_{j-1}/2^{c+1}$ candidates for node $B_j$ to pick and add to its $c$-th $k$-bucket, and node $B_i$ is in this candidate set. We can consider two possible scenarios. When $N_{share}(c) > k$, node $B_j$ randomly picks $k$ nodes from the candidate set to put in its routing table; when $N_{share}(c) \leq k$, all the nodes in the candidate set will be chosen. Therefore, the probability of node $B_j$ adding node $B_i$ into its routing table is

$$P_{add}(c) = \begin{cases} \frac{k}{\frac{N_{j-1}}{2^{c+1}}}, & \frac{N_{j-1}}{2^{c+1}} > k \\ 1, & \frac{N_{j-1}}{2^{c+1}} \leq k \end{cases} \tag{8.8}$$

Let $c_1 = \lceil \log^{N_{j-1}/k} - 1 \rceil$, i.e., $N_{j-1}/2^{c_1+1} = k$, we can rewrite Eq. (8.8) and get Eq. (8.9).

$$P_{add}(c) = \begin{cases} \frac{k}{\frac{N_{j-1}}{2^{c+1}}}, & c < c_1 \\ 1, & c \geq c_1 \end{cases} \tag{8.9}$$

Therefore, the number of $Nodes_{after}$ that would have node $B_i$ in its routing table can be calculated as follows:

$$N_{after}(i) = \sum_{c=0}^{m-1} P_{share}(c) \times P_{add}(c) \tag{8.10}$$

For a specific node $B_i$, the total number of nodes having it in their routing tables is

$$N_{routing}(i) = N_{before}(i) + N_{after}(i) \tag{8.11}$$

and the average number of nodes that have a monitoring node in their routing tables is

$$\overline{N}_{routing} = \frac{1}{N} \sum_{i=1}^{N} N_{routing}(i) \tag{8.12}$$

#### 8.3.4.2 Simulation Evaluation

Still we simulate the same Kademlia-based P2P botnet as the one in Sects. 8.3.2.4 and 8.3.3.4. We carry out two types of experiments: P2P botnets without churn and botnets with churn, where churn refers to the network dynamics caused by nodes' joining and leaving activities.

For P2P botnets without churn, we consider once a botnet is constructed, the botnet is stable, i.e., no nodes will leave the network and no more nodes will join the network as well. $\overline{N}_{routing}$, the average number of nodes which have a given monitoring node in their routing tables for different scales of networks is shown in Fig. 8.7a. We can see that our analysis precisely estimates $\overline{N}_{routing}$.

However, in the real world, the churn does exist in P2P botnets. To make our experiment more realistic, we introduce the churn events in our simulations. In our simulated P2P network, node joining and node leaving events will happen, and we assume the time interval between two churn events $t_{churn}$ follows a truncated normal distribution (i.e., $t_{churn} \sim N(\mu, \sigma^2)$ and $t_{churn} > 0$). In order not to favor any one of the node's joining and node's leaving, when a churn event happens, we set the probability of it being a node's joining $P_{in}$ and of it being a node's leaving $P_{out}$ to be the same, i.e., they are both 50 %.

Figure 8.7b shows our simulation results when considering churn. In our experiments, all simulations run for the same amount of time (30,000 unit time) and $t_{churn}$ follows the same distribution ($\mu = 15$ and $\sigma = \mu/3$). As a result, in each simulation, there are around 2,000 node joining/leaving events. If the size of the network is small, only a small fraction of original bot nodes (e.g., the first $N$ nodes) still exist in the network when the simulation ends. But in a relatively large network, a large fraction of original nodes still exist in the network. For example, when $N = 200$, there are around 4–5 % of the first 200 nodes still in the network at the end of the simulation; while when $N$=30,000, 96 % of the original 30,000 nodes remain in the network. From another perspective, we can view this phenomenon as the illustration of monitoring performance under different churn intensities. Since in our experiments, we cover the sizes of network ranging from 200 to 30,000, we have considered the monitoring performance under different churn intensities. As what is shown in Fig. 8.7b, our analysis can still well evaluate $\overline{N}_{routing}$ even with churn.

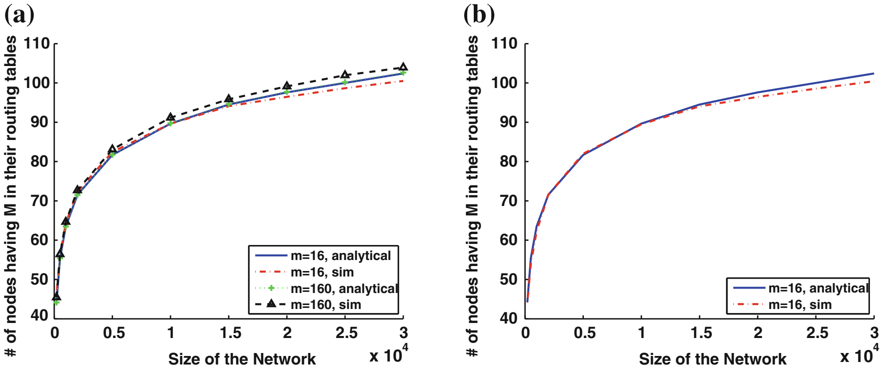


**Fig. 8.7** Comparison between analytical and simulation results of $\overline{N}_{routing}$. The simulated P2P botnet is Kademlia-based with a $D(1, 1, 8)$ routing table structure. **a** Without churn. **b** With churn

## 8.3.5 Others Countermeasures

In the following, we present several general ideas to defend against P2P botnets.

### 8.3.5.1 Detection

Being able to detect bot infection can stop a new-born botnet in its infant stage. Signature-based malware detection is effective and still widely used. But anti-signature techniques, such as polymorphic technique [31], make it possible for malware to evade such detection systems. Therefore, instead of doing static analysis, defenders start considering dynamic information for detection. For example, the system proposed by Gu et al. [21] is based on dynamic pattern matching.

Anomaly detection is another direction, since bots usually exhibit different behaviors from legitimate P2P users, such as sending queries periodically, always querying for the same content, or repeatedly querying but never downloading.

In addition, distributed detection is another approach, such as the self-defense infrastructure presented in [69], and two approaches against ultra-fast topological worms in [67].

### 8.3.5.2 Monitoring

Monitoring botnets help people better understand their motivations, working patterns, evolution of designs, etc. There are two effective ways to conduct P2P botnet monitoring.

For parasite and leeching P2P botnets, we can choose legitimate nodes in the host P2P networks as sensors for botnet monitoring. Usually sensors are peers that play important roles in the network communication, such as ultrapeers in Gnutella networks, such that more information can be collected. In DHT-based P2P networks, the search path of a specific key is relatively fixed, even if the search starts at different nodes. So the sensor selection depends on the monitoring targets and routing algorithm implemented in the system.

Honeypot techniques [49] are widely used for botnet monitoring. The way to set up honeypots in a P2P botnet is similar to choosing sensors. The difference is that honeypots are hosts added to the network on purpose by defenders, while sensors are chosen from the nodes who are already in the network.

### 8.3.5.3 Mitigation

The ultimate purpose of studying botnets is to shut them down. We can either (1) remove discovered bots, or (2) shut down C&C channels of botnets.

A botnet that relies on bootstrapping for construction is vulnerable during its early stage. Isolating or shutting down bootstrap servers or the bots in the initial list that are hard-coded in bot code can effectively prevent a new-born botnet from growing into a real threat.

P2P botnets can also be shut down or partially disabled by removing bot members. There are two modes of bot removal: random and targeted. The former means disinfecting the host whenever it is identified as a bot. The latter means removing critical bots, such as the ones that are important for C&C communication, when we have the knowledge of the topology or C&C architecture of a P2P botnet. Two metrics to evaluate the effectiveness of targeted removal were proposed in [63].

Shutting down detected bots is slow in disabling a botnet and sometimes impossible to do (e.g., you have no control of infected machines abroad). So a more effective and feasible way is to interrupt botnet C&C communication such that bots cannot receive orders from their botmaster. This approach has been carried out well for centralized botnets through shutting down the central C&C sites, but is believed to be more difficult to do for P2P botnets.

However, we find that this general understanding of “P2P botnet is much more robust against defense” is misleading. In fact, index-based P2P botnets are as vulnerable as centralized botnets, if the counter defense methods we presented in Sects. 8.3.2.2 and 8.3.3.2 are not implemented. Index poisoning defense (Sect. 8.3.2) and Sybil defense (Sect. 8.3.3) can be quite effective to fight against such botnets.

### *8.3.6 Discussion*

It is worthy to point out that the search process we discussed in Sects. 8.3.2.3 and 8.3.3.3 can be performed in two different manners—iterative and recursive. Let us use the scenario presented in Fig. 8.3 to explain the difference between these two search modes: the iterative search route would be $A \rightarrow B_1 \rightarrow A \rightarrow B_2 \rightarrow A \rightarrow B_3 \rightarrow A \rightarrow B_4 \rightarrow A \rightarrow B_5 \rightarrow A \rightarrow D$, while the recursive search route would be $A \rightarrow B_1 \rightarrow B_2 \rightarrow B_3 \rightarrow B_4 \rightarrow B_5 \rightarrow D \rightarrow B_5 \rightarrow B_4 \rightarrow B_3 \rightarrow B_2 \rightarrow B_1 \rightarrow A$. A P2P protocol could use either one of them. For instance, Kademlia employs the iterative search algorithm, and the Nugache P2P botnet has implemented the recursive routing. Although the routes are different, our analysis applies to both of the search algorithms. This is because, in our analysis, what we care about is the number of distinct nodes that a search message would go through besides node $A$ and node $D$ within a target zone; this number depends on the length of the path $A \rightarrow B_1 \rightarrow B_2 \rightarrow B_3 \rightarrow B_4 \rightarrow B_5 \rightarrow D$, and in both search cases this length is the same.

In addition, as we mentioned before, the ideas of index poisoning and Sybil were first introduced in legitimate P2P networks, and passive monitoring can also be deployed in current P2P file sharing networks. When P2P botnets use the same P2P protocols, these techniques can be leveraged to fight against these botnets as well. Therefore, our analysis of these three techniques is applicable to not only P2P botnets, but also to legitimate P2P systems. Moreover, in our analysis, we mainly talked about P2P botnets utilizing Kademlia for command and control; however, index poisoning defense and Sybil defense techniques are also valid for P2P botnets that rely on P2P networks for other communication, such as Storm botnet, which utilizes a P2P network to help bots join its hierarchical multi-tier command and control network. Therefore, our analysis is valid for general P2P botnets, no matter whether they use P2P networks for command dissemination, or for other communications.

## 8.4 Related Work

P2P botnets, as a new form of botnet, have appeared in the last few years and obtained people's attention. In [19] Grizzard et al., conducted a case study on Trojan.Peacomm botnet. Later on, Holz et al., adapted tracking technique used to mitigate IRC-based botnets and extended it to analyze Storm worm botnets [27]. Trojan.Peacomm botnet and Stormnet are two typical P2P botnets. Although bots in these two botnets are infected by two different malware, Trojan.Peacomm and Storm worm respectively, both of their C&C mechanisms are based on Kademlia [39]. And a botnet protocol which is also based on Kademlia was proposed by Starnberger et al. [50]. Moreover, to be well prepared for the future, there are some other botnets whose architecture is similar to P2P architecture, such as an advanced hybrid P2P botnet [63], super botnet [61] and the Loosely Coupled P2P botnet (lcbot) [11]. Wang et al. [62] studied P2P botnets along multiple dimensions including botnet construction, command and control mechanisms, performance measurements, and mitigation approaches. Rossow et al. [45] used formal graph model to capture the intrinsic properties and fundamental vulnerabilities of P2P botnets; however, this work does not provide mathematical modeling of the mitigation techniques against P2P botnets. Han et al. [25] presented a matrix model for P2P botnets and provided formulas of five performance metrics including connection degree, connection degree ratio, connection ratio, exposure ratio and average hop count. Singh et al. [48] Built a distributed intrusion detection framework that can be used to detect P2P Botnet by machine learning approach.

There have been some systematic studies on general botnets. Barfor and Yegneswaran [7] studied and compared four widely-used IRC-based botnets from seven aspects: botnet control mechanisms, host control mechanisms, propagation mechanisms, exploits, delivery mechanisms, obfuscation and deception mechanisms. Considering aspects such as attacking behavior, C&C model, rally mechanism, communication protocol, evasion technique and other observable activities, Trend Micro [40] proposed a taxonomy of botnet threads. In [13] Dagon et al., also presented a taxonomy for botnets but from a different perspective. Their taxonomy focuses on the botnet structure and utility. And in 2008, a botnet research survey [70] done by Zhu et al., classified research work on botnets into three categories: bot anatomy, wide-area measurement study and botnet modeling and future botnet prediction. Bailey et al. presented another survey, which provided an overview of current botnets, discussed how different types of networks can affect the effectiveness of botnet detection mechanism, and talked about various detection techniques that have been proposed [6]. What differs our work from theirs is that we focused on newly appeared P2P botnets, and tried to understand P2P botnets along four dimensions: P2P botnet construction, C&C mechanisms, measurements and defenses.

Modeling P2P botnet propagation is one dimension we did not discuss in this chapter. Król [34] presented theoretical study of malware propagation in various complex networks. In the research work [46], Ruitenbeek and Sanders presented a stochastic model of the creation of a P2P botnet. In [14], Dagon et al., proposed a diurnal propagation model for computer online/offline behaviors and showed that regional bias in infection will affect the overall growth of the botnet. [44] formulated an analytical model that emulates the mechanics of a decentralized Gnutella type of peer network and studied the spread of malware on such networks. Both [68] and [59] presented an analytical propagation model of P2P worms, but the former targets topological scan based P2P worms, while the latter targets passive scan based P2P worms.

Many researchers have investigated on detection and mitigation of traditional centralized C&C botnets. Wurzinger et al. presented an approach to automatically generate models for botnet detection [66]. Their models are generated based on the fact that every bot responds to the botmaster in a specific way. Researchers try to distinguish bot behavior from human behavior, in order to detect botnets. For example, in [38], malicious channels created by bots are differentiated from normal traffic generated by human beings; and in [22], hypothesis testing is used to separate botnet C&C dialogs from human-human conversations. Pattern recognition approaches and clustering algorithms are widely used for botnet detection. Chang and Daniels proposed a node behavior profiling approach to capture the node behavior clusters in a network for botnet C&C communication detection [10]. And a Bayesian approach for detecting bots based on the similarity of their DNS traffic to that of known bots is presented in [60]. In addition, Gu et al., proposed three botnet detection systems: BotMiner [20]—a botnet detection framework by performing cross cluster correlation on captured communication and malicious traffic, BotSniffer [23]—a system that can identify botnet C&C channels in a local area network based on the observation that bots within the same botnet will demonstrate spatial-temporal correlation and similarity and BotHunter [21]—a bot detection system using IDS-Driven Dialog Correlation according to defined bot infection dialog model.

Furthermore, botnet infiltration and monitoring is also an very active topic in botnet research community. In [29], Kang et al., presented a passive P2P monitor, which can enumerate the infected hosts regardless whether or not they are behind a firewall or NAT, and conducted an empirical study on Storm botnet. Li et al. [35], monitored botnets probing activities and addressed the problems like botnet's scanning strategies and attack target selection policies. In [30], Kanich et al., pointed out a number of challenges that arise in using crawling to measure the size, topology, and dynamism of distributed botnets. People infiltrate specific botnets, such as MegaD botnet [12], Torpig bot [56] and [41], in order to understand their architectures, communication protocols, behaviors, etc. In addition, botnet infiltration and monitoring can be very helpful for fighting against malicious activities. In [32, 33, 42], the data collected through infiltrating and monitoring botnets are used for spam detection and analysis.

Some researchers have studied theoretical models of complex network in terms of network robustness against general network failure or malicious attacks. Schneider et al. [47] presented mathematical analysis of complex networks and introduced a new measure for robustness. They have demonstrated that electricity grid and Internet can significantly improve their robustness against malicious attacks with small changes in the network structure. Hayes et al. [26] presented a new algorithm to improve self-healing in peer-to-peer networks against node insertion or deletion attacks. Louzada et al. [37] presented a new rewiring method to modify a network topology improving its robustness, based on the evolution of the network largest component during a sequence of targeted attacks.

## 8.5 Conclusion

P2P botnets, as a new advanced form of botnets, have attracted attentions from both botmasters and security defenders. In this chapter, we first presented a systematical study on P2P botnets. We discussed in detail each stage in the life cycle of P2P botnets, and classified P2P botnets into three categories: parasite, leeching and bot-only P2P botnets. Then among possible directions for P2P botnet defense, we focused on two mitigation techniques against P2P botnets—index poisoning defense and Sybil defense, and one monitoring technique— passive monitoring, and analyzed their effectiveness in terms of several factors, such as the size of a botnet, the settings of the communication protocol, the range of the defense deployment. Simulation-based experiments have shown that our analysis is accurate. This work provides guidance for security professionals on how to carry out these three defense techniques to achieve better performance. In the mean time, we discussed how attackers might react to avoid or reduce the effectiveness of index poisoning defense and Sybil defense techniques, which help people get prepared for the future in case such methods are deployed by attackers. Furthermore, based on our study, we obtained a counterintuitive finding: because of the similar information dissemination structure, P2P botnets that rely on index for command or other critical information dissemination may be as easy (or as hard) to be shut down as the centralized botnets.

## References

1. http://www.symantec.com/security_response/index.jsp
2. http://en.wikipedia.org/wiki/Kad_network
3. emule. http://www.emule-project.net/
4. SdDrop. http://www.viruslist.com/en/viruses/encyclopedia?virusid=24282
5. Andriesse, D., Rossow, C., Stone-Gross, B., Plohmann, D., Bos, H.: Highly resilient peer-to-peer botnets are here: an analysis of Gameover Zeus. In: Proceeding of 8th International Conference on Malicious and Unwanted Software: “The Americas” (MALWARE) (2013)

6. Bailey, M., Cooke, E., Jahanian, F., Xu, Y., Karir, M.: A survey of botnet technology and defenses. In: Proceeding of the 2009 Cybersecurity Applications and Technology Conference for Homeland Security (2009)
7. Barford, P., Yegneswaran, V.: An Inside Look at Botnets. Advances in Information Security, In Series (2006)
8. Baumgart, I., Heep, B., Krause, S.: OverSim: a flexible overlay network simulation framework. In: Proceedings of the 10th IEEE Global Internet Symposium (GI'07) in Conjunction with IEEE INFOCOM'07, Anchorage, AK (2007)
9. Bhaduri, K., Das, K., Kargupta, H.: Peer-to-peer data mining, privacy issues, and games. **4476**, 1–10 (2007)
10. Chang, S., Daniels, T.E.: P2P botnet detection using behavior clustering and statistical tests. In: Proceedings of the 2nd ACM workshop on Security and Artificial Intelligence (AISec'09), Chicago (2009)
11. Chang, S., Zhang, L., Guan, Y., Daniels, T.E.: A framework for P2P botnets. In: Proceedings of the 2009 International Conference on Communications and Mobile Computing (CMC'09), Kunming,Yunnan, China (2009)
12. Chox, C.Y., Caballeroyx, J., Grierx, C., Paxsonzx, V., Song, D.: Insights from the inside: a view of botnet management from infiltration. In: Proceedings of the 3rd USENIX Workshop on Large-Scale Exploits and Emergent Threats, San Jose (2010)
13. Dagon, D., Gu, G., Lee, C., Lee, W.: A taxonomy of botnet structures. In: Proceedings of the 23rd Annual Computer Security Applications Conference (ACSAC'07) (2007)
14. Dagon, D., Zou, C.C., Lee, W.: Modeling botnet propagation using time zones. In: Proceedings of the 13th Annual Network and Distributed System Security Symposium (NDSS'06) (2006)
15. Damfling, H.: Gnutella web caching system. http://www.gnucleus.org/gwebcache/specs.html
16. Davis, C.R., Fernandez, J.M., Neville, S., McHugh, J.: Sybil attacks as a mitigation strategy against the storm botnet. In: Proceedings of the 3rd International Conference on Malicious and Unwanted Software (Malware'08) (2008)
17. Douceur, J.R.: The sybil attack. In: Proceedings of the 1st International Workshop on Peer-to-Peer Systems (2002)
18. Enright, B., Voelker, G., Savage, S., Kanich, C., Levchenko, K.: Storm: when researchers collide. USENIX Login 33(4) (2008)
19. Grizzard, J.B., Sharma, V., Nunnery, C., Kang, B.B., Dagon, D.: Peer-to-Peer botnets: overview and case study. In: Proceedings of the 1st USENIX Workshop on Hot Topics in Understanding Botnets (HotBots'07), Cambridge, MA (2007)
20. Gu, G., Perdisci, R., Zhang, J., Lee, W.: BotMiner: clustering analysis of network traffic for protocol—and structure-independent botnet detection. In: Proceedings of the 17th USENIX Security Symposium (Security'08) (2008)
21. Gu, G., Porras, P., Yegneswaran, V., Fong, M., Lee, W.: BotHunter: Detecting malware infection through IDS-driven dialog correlation. In: Proceedings of the 16th USENIX Security Symposium (Security'07) (2007)
22. Gu, G., Yegneswaran, V., Porras, P., Stoll, J., Lee, W.: Active botnet probing to identify obscure command and control channels. In: Proceedings of the Annual Computer Security Applications Conference (ACSAC'09), Hawaii (2009)
23. Gu, G., Zhang, J., Lee, W.: BotSniffer: detecting botnet command and control channels in network traffic. In: Proceedings of the 15th Annual Network and Distributed System Security Symposium (NDSS'08) (2008)
24. Ha, D.T., Yan, G., Eidenbenz, S., Ngo, H.Q.: On the effectiveness of structural detection and defense against P2P-based botnets. In: Proceedings of the 39th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN'09), Estoril, Lisbon, Portugal (2009)
25. Han, Q., Yu, W., Zhang, Y., Zhao, Z.: Modeling and evaluating of typical advanced peer-to-peer botnet. Perform. Eval. **72**, 1–15 (2014)
26. Hayes, T.P., Saia, J., Trehan, A.: The forgiving graph: a distributed data structure for low stretch under adversarial attack. Distrib. Comput. **25**(4), 261–278 (2012)

27. Holz, T., Steiner, M., Dahl, F., Biersack, E.W., Freiling, F.: Measurements and mitigation of peer-to-peer-based botnets: a case study on storm worm. In: Proceedings of the 1st Usenix Workshop on Large-scale Exploits and Emergent Threats (LEET), San Francisco,USA (2008)
28. Jelasity, M., Bilicki, V.: Towards automated detection of peer-to-peer botnets: on the limits of local approaches. In: Proceedings of the 2nd USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET'09), Boston, MA (2009)
29. Kang, B.B., Chan-Tin, E., Lee, C.P., Tyra, J., Kang, H.J., Nunnery, C., Wadler, Z., Sinclair, G., Hopper, N., Dagon, D., Kim, Y.: Towards complete node enumeration in a peer-to-peer botnet. In: Proceedings of the 2009 ACM Symposium on Information, Computer and Communications Security (ASIACCS), Sydney, Australia (2009)
30. Kanich, C., Levchenko, K., Enright, B., Voelker, G.M., Savage, S.: The heisenbot uncertainty problem: challenges in separating bots from chaff. In: Proceedings of the USENIX Workshop on Large-Scale Exploits and Emergent Threats, San Franciso, CA (2008)
31. Kolesnikov, O., Dagon, D., Lee, W.: Advanced polymorphic worms: evading IDS by blending in with normal traffic. Technical Report, Georgia Technology, Georgia (2004–2005)
32. Kreibich, C., Kanich, C., Levchenko, K., Enright, B., Voelker, G.M., Paxson, V., Savage, S.: On the spam campaign trail. In: Proceedings of the USENIX Workshop on Large-Scale Exploits and Emergent Threats, San Franciso, CA (2008)
33. Kreibich, C., Kanich, C., Levchenko, K., Enright, B., Voelker, G.M., Paxson, V., Savage, S.: Spamcraft: an inside look at spam campaign orchestration. In: Proceedings of the USENIX Workshop on Large-Scale Exploits and Emergent Threats, Boston, MA (2009)
34. Król, D.: Propagation phenomenon in complex networks: theory and practice. New Gener. Comput. **32**(3–4), 187–192 (2014)
35. Li, Z., Goyal, A., Chen, Y., Paxson, V.: Automating analysis of large-scale botnet probing events. In: Proceedings of ACM Symposium on Information, Computer and Communications Security (2009)
36. Liang, J., Naoumov, N., Ross, K.W.: The index poisoning attack in P2P file sharing systems. In: Proceedings of the Infocom, Barcelona (2006)
37. Louzada, V.H., Daolio, F., Herrmann, H.J., Tomassini, M.: Smart rewiring for network robustness. J. Complex Netw. **1**(2), 150–159 (2013)
38. Lu, W., Tavallaee, M., Ghorbani, A.A.: Automatic discovery of botnet communities on large-scale communication networks. In: Proceedings of the 2009 ACM Symposium on Information, Computer and Communications Security (ASIACCS), Sydney, Australia (2009)
39. Maymounkov, P., Mazieres, D.: Kademlia: a peer-to-peer information system based on the xor metric. In: Proceedings of the 1st International Workshop on Peer-to-Peer Systems, pp. 53–65 (2002)
40. Micro, T.: Taxonomy of botnet threats (2006)
41. Nunnery, C., Sinclair, G., Kang, B.B.: Tumbling down the rabbit hole: exploring the idiosyncrasies of botmaster systems in a multi-tier botnet infrastructure. In: Proceedings of the 3rd USENIX Workshop on Large-Scale Exploits and Emergent Threats, San Jose,CA (2010)
42. Pitsillidis, A., Levchenko, K., Kreibich, C., Kanich, C., Voelker, G.M., Paxson, V., Weaver, N., Savage, S.: Botnet judo: fighting spam with itself. In: Proceedings of the Network and Diestributed System Security Symposium (NDSS), San Diego, CA (2010)
43. Porras, P., Saidi, H., Yegneswaran, V.: A Multi-perspective analysis of the storm (Peacomm) Worm. Technical Report, SRI (2007)
44. Ramachandran, K., Sikdar, B.: Modeling malware propagation in Gnutella type peer-to-peer networks. In: Proceedings of the 20th International Parallel and Distributed Processing Symposium (IPDPS'06), Rhodes Island, Greece (2006)
45. Rossow, C., Andriesse, D., Werner, T., Stone-Gross, B., Plohmann, D., Dietrich, C.J., Bos, H.: SoK: P2PWNEDł modeling and evaluating the resilience of peer-to-peer botnets. In: Proceedings of 2013 IEEE Symposium on Security and Privacy (2013)
46. Ruitenbeek, E.V., Sanders, W.H.: Modeling peer-to-peer botnets. In: Proceedings of the 5th International Conference on Quantitative Evaluation of Systems (QEST'08), St Malo (2008)

47. Schneider, C.M., Moreira, A.A., Andrade, J.S., Havlin, S., Herrmann, H.J.: Mitigation of malicious attacks on networks. In: Proceedings Nat Acad Sci USA 108, p. 3838C3841 (2011)
48. Singh, K., Guntuku, S.C., Thakur, A., Hota, C.: Big data analytics framework for peer-to-peer botnet detection using random forests. Inf. Sci. (2014)
49. Spitzner, L.: Honeypots: Tracking Hackers. Addison-Wesley Longman Publishing Co., Inc, Boston (2002)
50. Starnberger, G., Kruegel, C., Kirda, E.: Overbot—a botnet protocol based on kademlia. In: Proceedings of the 4th International Conference on Security and Privacy in Communication Networks (SecureComm) (2008)
51. Steiner, M., En-Najjary, T., Biersack, E.W.: A global view of KAD. In: Proceedings of the ACM Internet Measurement Conf. (IMC), San Diego, USA (2007)
52. Steiner, M., En-Najjary, T., Biersack, E.W.: Exploiting KAD: possible uses and misuses **37**(5), 65–70 (2007)
53. Stewart, J.: Inside the storm: protocols and encryption of the storm botnet. Black Hat Conference, USA (2008) http://www.blackhat.com/presentations/bh-usa-08/Stewart/BH_US_08_Stewart_Protocols_of_the_Storm.pdf
54. Stock, B., Goel, J., Engelberth, M., Freiling, F.C.: Walowdac—analysis of a peer-to-peer botnet. In: Proceedings of the European Conference on Computer Network Defense (EC2ND'09) (2009)
55. Stoica, I., Morris, R., Karger, D., Kaashoek, M.F., Balakrishnan, H.: Chord: a scalable peer-to-peer lookup service for internet applications, AC. In: Proceedings of the ACM SIGCOMM, San Deigo, CA (2001)
56. Stone-Gross, B., Cova, M., Cavallaro, L., Gilbert, B., Szydlowski, M., Kemmerer, R., Kruegel, C., Vigna, G.: Your botnet is my botnet: analysis of a botnet takeover. In: Proceedings of the ACM CCS, Chicago, IL (2010)
57. Stover, S., Dittrich, D., Hernandez, J., Dietrich, S.: Analysis of the storm and nugache trojans: P2P is here. USENIX login **32**(6), 18–27 (2007)
58. Stutzbach, D., Rejaie, R.: Improving lookup performance over a widely-deployed DHT. In: Proceedings of the IEEE INFOCOM, Barcelona, Spain (2006)
59. Thommes, R., Coates, M.: Epidemiological modelling of peer-to-peer viruses and pollution. In: Proceedings of the IEEE Infocom, Barcelona (2006)
60. Villamarín-Salomón, R., Brustoloni, J.C.: Bayesian bot detection based on DNS traffic similarity. In: Proceedings of the the 24th Annual ACM Symposium on Applied Computing (SAC'09), Honolulu, Hawaii (2009)
61. Vogt, R., Aycock, J., Jacobson, M.: Army of botnets. In: Proceedings of the 2007 Network and Distributed System Security Symposium (NDSS) (2007)
62. Wang, P., Aslam, B., Zou, C.C.: Peer-to-peer botnets. In: Stavroulakis, P., Stamp, M. (eds.) Handbook of Information and Communication Security. Springer, New York (2010)
63. Wang, P., Sparks, S., Zou, C.C.: An advanced hybrid peer-to-peer botnet. In: Proceedings of the 1st USENIX Workshop on Hot Topics in Understanding Botnets (HotBots'07), Cambridge, MA (2007)
64. Wang, P., Tyra, J., Chan-Tin, E., Malchow, T., Kune, D.F., Hopper, N., Kim, Y.: Attacking the kad network. In: Proceedings of the 4th International Conference on Security and Privacy in Communication Netowrks (SecureComm'08) (2008)
65. Wang, P., Wu, L., Aslam, B., Zou, C.C.: A systematic study on peer-to-peer botnets. In: Proceedings of International Conference on Computer Communications and Networks (ICCCN) (2009)
66. Wurzinger, P., Bilge, L., Holz, T., Goebel, J., Kruegel, C., Kirda, E.: Automatically generating models for botnet detection. In: Proceedings of the 14th European Symposium on Research in Computer Security (ESORICS), Saint Malo, France (2009)
67. Xie, L., Zhu, S.: A feasibility study on defending against ultra-fast topological worms. In: Proceedings of the 7th IEEE International Conference on Peer-to-Peer Computing (P2P'07), Galway, Ireland (2007)

68. Yu, W., Boyer, P.C., Chellappan, S., Xuan, D.: Peer-to-peer system-based active worm attacks: modeling and analysis. In: Proceedings of the IEEE International Conference on Communications (ICC) (2005)
69. Zhou, L., Zhang, L., McSherry, F., Immorlica, N., Costa, M., Chien, S.: A first look at peer-to-peer worms: threats and defenses. In: Proceedings of the 4th International Workshop on Peer-to-Peer Systems (IPTPS'05) (2005)
70. Zhu, Z., Lu, G., Chen, Y., Fu, Z.J., Roberts, P., Han, K.: Botnet research survey. In: Proceedings of the 32nd Annual IEEE International Computer Software and Applications (COMPSAC'08) (2008)

# Chapter 9
# Generating Robust and Efficient Networks Under Targeted Attacks

**Vitor H.P. Louzada, Fabio Daolio, Hans J. Herrmann and Marco Tomassini**

**Abstract** Much of our commerce and travel depends on the efficient operation of large scale networks. Some of those, such as electric power grids, transportation systems, communication networks, and others, must maintain their efficiency even after several failures, or malicious attacks. We outline a procedure that modifies any given network to enhance its robustness, defined as the size of its largest connected component after a succession of attacks, whilst keeping a high efficiency, described in terms of the shortest paths among nodes. We also show that this generated set of networks is very similar to networks optimized for robustness in several aspects such as high assortativity and the presence of an onion-like structure.

## 9.1 Introduction

In recent years, insights provided by network analysis have attracted a lot of attention from practitioners. As a result, it has been shown that several artificial (e.g. the Internet, electric-grids, etc.) and natural systems (e.g. chemical reaction networks, food networks, gene regulatory networks, etc.) present characteristics that allows one to classify them as Complex Networks. Their structure and the dynamics of phenomena taking place on them have been intensively studied [5], thanks to the availability of large data sets [8].

An important aspect of a network is the capability to withstand failures and fluctuations in the functionality of its nodes and links. The design of networked infrastructures with these capabilities can be thought of as an optimization task. An

V.H.P. Louzada (✉) · H.J. Herrmann
Computational Physics, IfB, ETH Zurich, Wolfgang-Pauli-Strasse 27,
Zurich 8093, Switzerland
e-mail: louzada@ethz.ch

H.J. Herrmann
Departamento de Física, Universidade Federal Do Ceará,
Fortaleza, Ceará 60451-970, Brazil

F. Daolio · M. Tomassini
Faculty of Business and Economics, University of Lausanne, Lausanne, Switzerland

D. Król et al. (eds.), *Propagation Phenomena in Real World Networks*,
Intelligent Systems Reference Library 85, DOI 10.1007/978-3-319-15916-4_9

early important work in this field is Albert et al. [1] where the authors showed by numerical simulations that scale-free networks, while they are robust against random removal of nodes, are much more vulnerable to the removal of nodes according to their degree. In other words, in a scale-free network if the nodes are removed in decreasing order of degree, starting with the most connected ones, then the network falls apart very quickly.

In Schneider et al. [9], a procedure is described that successfully modifies scale-free networks so that the largest connected component still has a considerable size after several attacks targeted at the most connected nodes. This feature guarantees that there is at least one path connecting a large number of nodes after attacks and is considered an appropriate definition of robustness. A natural question that follows is the maintenance of network efficiency after attacks, i.e., a network is efficient in this sense if “good paths” among nodes do not cease to exist after several targeted failures. Using a consolidated definition of efficiency, we propose an optimization procedure that modifies existing networks in order to improve their efficiency under targeted attacks.

This chapter is organized as follows. In Sect. 9.2, we present our measures of robustness, efficiency, and a method to optimize a specific characteristic of a network. Then, we show in Sect. 9.3 several comparisons of optimized and unoptimized networks. We highlight the major points of our contribution in Sect. 9.4.

## 9.2 Model

The proposed methodology is an extension of the work of Schneider et al. [9], who used a hill-climbing procedure to optimize robustness against targeted attacks. We modify this approach by adding a simulated annealing strategy [4] to avoid the search getting trapped in local maxima. Previous approaches have successfully used simulated annealing to increase network robustness [3]. Here however we extend our focus to the following objectives: Robustness, Efficiency, and a combined measure of both. We create three sets of networks optimized for these cost functions and compare their characteristics. In what follows, we describe the cost functions and the optimization procedure.

### *9.2.1 Robustness*

The definition of network robustness might change according to a specific application. In this work, we call an attack the removal of a node of the network, and the robustness we measure by the size of the largest connected component (LCC) of the network after this removal, as proposed by Schneider et al. [9]. To quantify it, we proceed with a series of attacks and subsequently measure the robustness after each node removal. Hence, robustness $R$ is defined as:

$$R = \frac{1}{N} \sum_{Q=1}^{N} S\left(\frac{Q}{N}\right) \tag{9.1}$$

where $N$ is the number of nodes, $Q$ is the number of nodes removed from the network, and $S(q)$ is the size of the LCC after a fraction $q = Q/N$ of nodes were removed. The attacks performed are targeted to the nodes with highest degree of the network: we find the most connected node, remove it, calculate $S(q)$, update the degrees, and find the new most connected node to repeat the process. In case two nodes have the same degree, we choose the one with the smallest index. The value $R$ is therefore unique for each network.

### *9.2.2 Efficiency*

One can think of network efficiency as a low cost of communication among its members. In this light, we relate efficiency with the shortest paths between all pairs of nodes, thus following Latora and Marchiori [6] who defined the network efficiency $E$ as:

$$E = \sum_{\substack{i,j=1 \\ i \neq j}}^{N} \frac{1}{l_{ij}} \tag{9.2}$$

where $l_{ij}$ stands for the shortest path length between nodes $i$ and $j$. If $i$ and $j$ belong to separate connected components of the network, we set $l_{ij} \to \infty$ to guarantee a consistent behavior of the cost function.

### *9.2.3 Integral Efficiency*

Keeping in mind that we would like to keep the efficiency of networks after attacks, it is straightforward to modify the definition of $E$ to account for this. Hence, we define Integral Efficiency $IntE$ as:

$$IntE = \frac{1}{N} \sum_{Q=1}^{N} E\left(\frac{Q}{N}\right) \tag{9.3}$$

where $E(q)$ stands for the efficiency of the network after the removal of $q = Q/N$ nodes. Nodes are removed according to a targeted attack such as in Sect. 9.2.1. The value of $E(0)$ is the cost function $E$ defined in Sect. 9.2.2. By choosing this quantity instead of $E$, which does not consider nodes removal in its definition, we try to avoid that the shortest paths among nodes increases after targeted attacks.

### 9.2.4 Optimization Procedure

In their work, Schneider et al. [9] propose a simple hill-climbing search to modify the network topology in order to optimize the robustness $R$ whilst keeping the degree of each node fixed. This restriction in often present in the modification of artificial systems, such as electric grids where constructing a receiver for a new power line in a station might be impractical. Hence, only swaps between lines (edges in the network) are possible. A consequence of this restriction is that the underlying degree distribution of the network remains unchanged after swaps. Clearly, if we had no constraints on the degree distribution, we could design the topology starting from scratch with the robustness and efficiency as objectives in mind, obtaining different optimal topologies.

Next, we present an improved version of the optimization approach using simulated annealing and we describe it for any cost function $M$ that changes after link modification:

1. **Initial State**. Let $G(N, E)$ be a network with $|N|$ nodes and $|E|$ edges.
2. **Edge swap**. Choose two pairs of edges $(i, j)$ and $(k, l) \in E$ randomly and create the network $G^*$ by deleting the edges $(i, j)$ and $(k, l)$, and adding the edges $(i, l)$ and $(k, j)$.
3. **Acceptance probability**. Calculate the transition probability $p$ of the system as:

$$p = \begin{cases} \exp\left(-\dfrac{M(G) - M(G^*)}{T}\right) & \text{if } M(G^*) < M(G) \\ 1 & \text{if } M(G^*) \geq M(G) \end{cases}$$

4. **Comparison**. Make $G = G^*$ with probability $p$, otherwise discard $G^*$. Return to Step 2.

This approach allows a network $G^*$ with $M(G^*) < M(G)$ to be chosen with finite probability. By doing this, global minima could be reached and inferior local minima could be avoided. Notice that, for the three cost functions studied here, the value of $M(G)$ is unique for each network $G$. Furthermore, by decreasing the value of $T$ according to the amount of edge swaps executed, it is possible to decrease the acceptance ratio of worst networks when an optimum point is close. We decrease the temperature as function of the number $\tau$ of edge swaps, by following the equation: $T(\tau) = 0.0001 \times 0.8^{\tau}$. Variations to this function have shown little effect on the results. The search is stopped when a predefined amount of edge swaps is reached.

## 9.3 Results

The procedure outlined in Sect. 9.2.4 is applied to the cost functions: $R$ (Robustness as described in Sect. 9.2.1), $E$ (Efficiency as described in Sect. 9.2.2), and $IntE$ (Integral Efficiency as described in Sect. 9.2.3), starting from the same set of randomly generated of Barabási-Albert (BA) networks. Hence, we created three sets of networks: *Robustness set*, *Efficiency set*, and *Integral Efficiency set*. As a control, we compare to the original set of BA networks, from now on called the *Unoptimized set*.

The Unoptimized set is composed of 100 networks of $n = 1000$ nodes and average degree $\langle k \rangle = 5.95$. The size of the networks was chosen based on a trade-off between the appearance of topological features such as the scale-free phenomenon, only present in large networks, and computational cost, as the $IntE$ cost function requires $O(n^3)$ operations to be calculated. The amount of edge swaps, 10,000, was chosen so that for each optimized set its cost function is already statistically different from the Unoptimized set. It is possible to see that this goal was achieved by comparing the values in bold for columns $\langle E \rangle$, $\langle R \rangle$, and $\langle IntE \rangle$ in Table 9.1. To provide a visualization of the network structure created, some examples of each set are drawn in Fig. 9.1.

To analyze the robustness of each set, a plot of $s(Q)$ versus $Q$ is shown in Fig. 9.2, in which the area below each curve represents $R$ for each set. As expected, the Robustness set shows a bigger area (23 % of increase), keeping a considerable size of the LCC after several attacks. Indeed, Schneider et al. [9] obtained an improvement of almost 75 % for this cost function, but by using a much more exhaustive approach: their search stops after 10,000 edge-swaps without increase in $R$. Therefore, our results show that it is possible to increase network robustness using less computational effort. The plot also shows that $E$, a cost function that does not consider attacks in its formulation, has a bad performance in this scenario. We conclude that, though more efficient, networks optimized exclusively for $E$ might not be appropriated in a realistic context, in contrast to $IntE$, which considers both effects. Moreover, it is interesting to note also that the curves for $R$ and the Integral Efficiency set have comparable areas, considering the standard deviation of the measurements as detailed in Table 9.1.

In Fig. 9.3, the cost function $IntE$ is analyzed through the plot of $E(Q)$ versus $Q$, showing that, as expected, the Integral Efficiency set has the better performance, i.e. the area under the corresponding curve is bigger. Interestingly, the curve referring to the set of networks obtained by optimizing for $E$ alone shows that both have about the same performance as the unoptimized ones for this cost function (data on Column $\langle IntE \rangle$ of Table 9.1).

**Table 9.1** Average values of the cost functions, standard deviation in subscripts

| Network set | $\langle k \rangle$ | $\langle cc \rangle$ | $\langle r \rangle$ | $\langle E \rangle$ | $\langle R \rangle$ | $\langle IntE \rangle$ |
|---|---|---|---|---|---|---|
| Unoptimized | $5.95_0$ | $0.0242_{0.0033}$ | $-0.085_{0.015}$ | $\mathbf{0.1486_{0.0012}}$ | $\mathbf{0.1837_{0.0053}}$ | $\mathbf{0.0308_{0.0011}}$ |
| E | $5.95_0$ | $0.0053_{0.0014}$ | $-0.076_{0.011}$ | $\mathbf{0.1539_{0.0015}}$ | $0.1826_{0.0056}$ | $0.0310_{0.0012}$ |
| R | $5.95_0$ | $0.0200_{0.0027}$ | $0.038_{0.024}$ | $0.1459_{0.0013}$ | $\mathbf{0.2266_{0.0055}}$ | $0.0372_{0.0012}$ |
| IntE | $5.95_0$ | $0.0195_{0.0029}$ | $0.055_{0.026}$ | $0.1456_{0.0013}$ | $0.2268_{0.0052}$ | $\mathbf{0.0391_{0.0012}}$ |

Each set comprises 100 networks with $n = 1000$ nodes. $\langle k \rangle$ = average degree, $\langle cc \rangle$ = average of clustering coefficient, $\langle r \rangle$ = average assortativity coefficient, $\langle E \rangle$ = average efficiency, $\langle R \rangle$ = average robustness, $\langle IntE \rangle$ = average integral efficiency

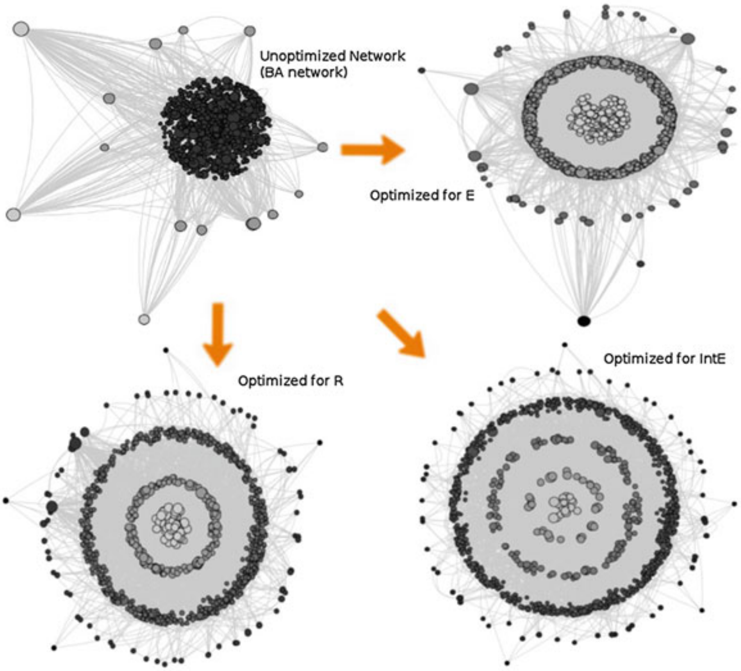


**Fig. 9.1** Examples of networks belonging to each set. Networks are drawn using the k-core decomposition, represented by the different intensities of *gray*

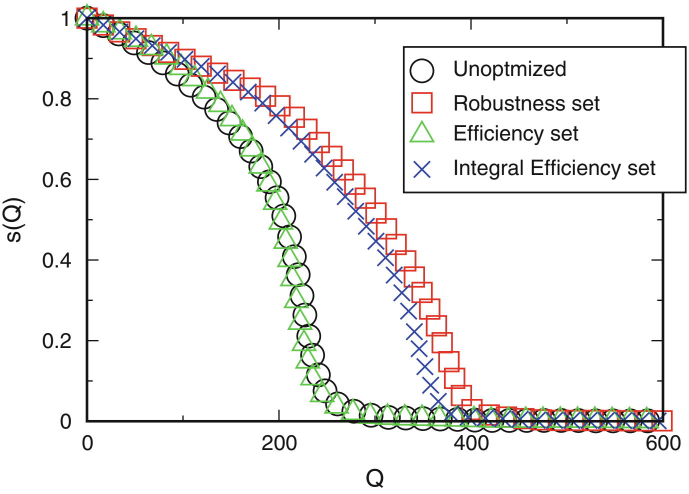


**Fig. 9.2** Largest component size after the removal of $Q$ nodes. The area bellow each *curve* is the cost function $R$. Symbols represent sets optimized for different cost functions and are larger than the standard deviation

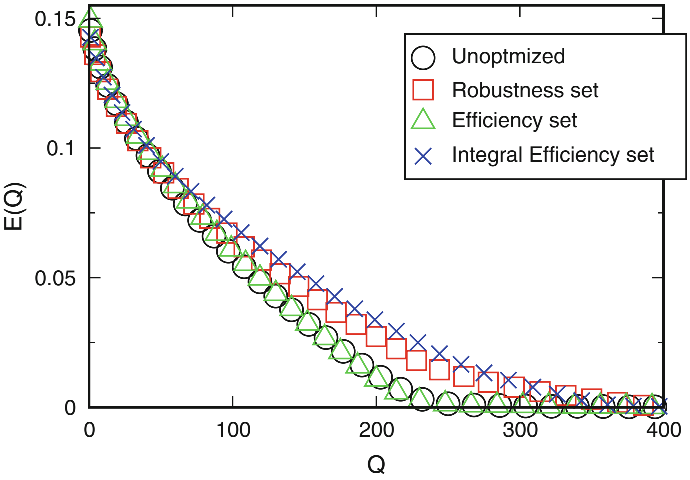


**Fig. 9.3** Network efficiency $E(Q)$ after the removal of $Q$ nodes. The area bellow each *curve* is the cost function $IntE$. Symbols represent sets optimized for different cost functions are larger than the standard deviation

Another interesting aspect of the work of Schneider et al. [9] is the topology obtained by this optimization: a so-called onion-like structure. In this topology, each layer is composed of nodes connected with nodes of the same degree, with few connections between layers. A direct procedure to generate this topology can be found in the work of Wu et al. [10].

To investigate the presence of an onion-like structure on our optimized sets, three quantities were analyzed. In Fig. 9.4, we show the *k-core* decomposition [2] for several $k$, showing that the Robustness and Integral Efficiency sets have several k-core's or layers, thus confirming a hierarchical structure of the network. The Efficiency set does not present this clear hierarchy, but has more layers than the Unoptimized set. In the inset of Fig. 9.4 we show that the Integral Efficiency set and the Robustness set of networks have the greater assortativity through the plot of Newman's $r$ coefficient [7]; the Efficiency set is as dissortative as the Unoptimized set.

Finally, we also measure the robustness for each layer of a network. To do so, we analyze the sub-graph of each network composed of $N_k$ nodes with degree smaller of equal to $k$. In this sub-graph, $S_k$ represents the size of its largest cluster. In Fig. 9.5, we plot $S_k/N_k$ for several values of $k$. This plot shows that the Robustness and the Integral Efficiency sets present practically the same increase in robustness with respect to the Unoptimized set. In contrast, the Efficiency set does not show any improvement with respect to the original scale-free unoptimized networks.

Given the several layers showed by the k-core decomposition, its dissortative nature, and the increase in robustness of each layer, we conclude that the Integral Efficiency set also has an onion-like structure similar to the Robustness set.

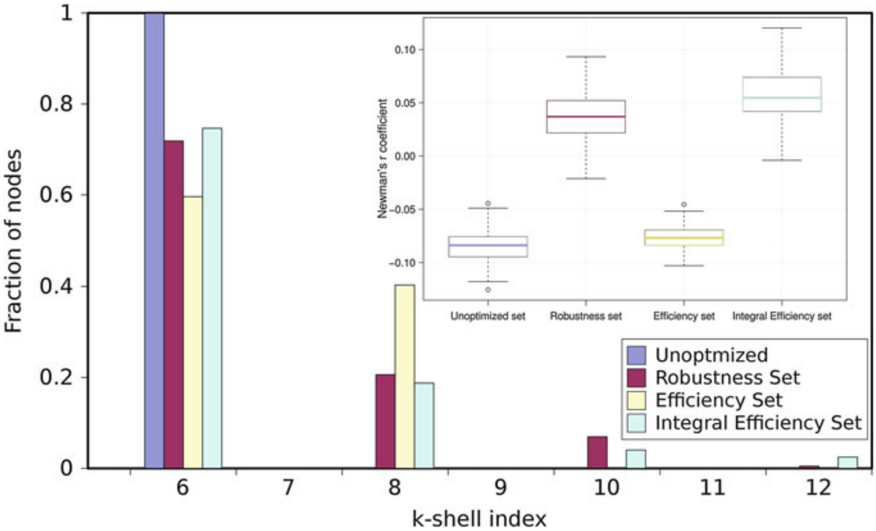


**Fig. 9.4** Main plot shows the K-core decomposition for several values of $k$. It can be seen that the same network optimized for $IntE$ presents more layers than the network resulted after the optimization for $R$. Inset shows *Box*-and-whiskers plot of the degree assortativity through Newman's r coefficient. *Thick* lines depict the median value; *lower* and *higher* hinges gives the 0.25 and 0.75 quantiles, respectively; the whiskers extend to 1.5 times this inter-quantile range. Values outside this range are considered outliers and appear as *circle dots* in the plot

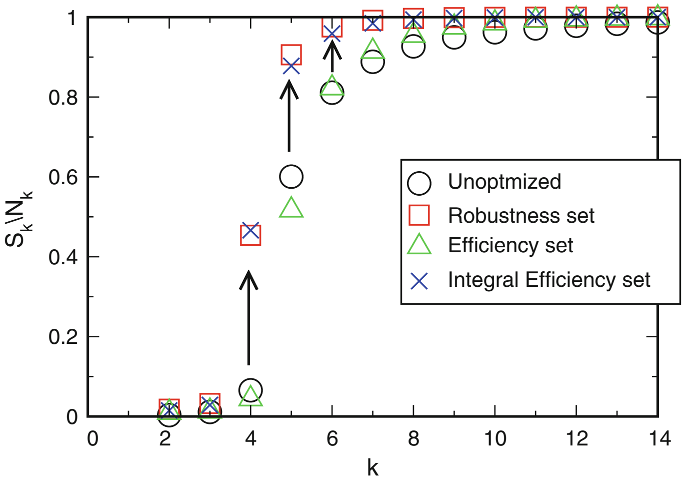


**Fig. 9.5** Relative size of the largest component in networks composed of nodes of degree less than $k$. Symbols represent sets optimized for different cost functions are larger than the standard deviation

## 9.4 Discussion

We outline here a procedure that optimizes a specific characteristic in any type of network and create three sets of BA networks with distinguishable features. Though BA networks are known to be resilient to random removals of nodes and present other interesting properties [1], we show here a method that creates networks with a certain specific characteristic enhanced, which might be useful in some realistic scenarios.

Firstly, our results show that the Integral Efficiency set substantially improved efficiency after attacks, compared to the Robustness, Efficiency, and Unoptimized sets. Moreover, this set also sustains a large connected cluster after attacks. Therefore, this cost-function could be used to generate highly robust and efficient networks.

Another important result of our work is that networks optimized for $IntE$ also present an onion-like structure. This result suggests that this structure is generically the optimal scale-free net independently of the chosen cost function. It also helps the design of networks from scratch, as it is possible to construct scale-free networks which present this structure.

It is also interesting to note that the Integral Efficiency set maintains several similarities with the Robustness set, such as: high assortativity, size of the largest cluster after attacks, efficiency after attacks, size of the largest cluster for each degree layer, and a hierarchical structure regarding the k-core decomposition. In fact, the Integral Efficiency set has a slightly better performance on assortativity and efficiency after attacks, while the Robustness set has a better performance on the others.

Future works might focus on the structures of the three generated sets. The Efficiency set does not present an onion-like structure, remaining unclear if this optimization could lead to a different structure. The Integral Efficiency set might have a hidden feature that differentiates it from the Robustness set. By finding a typical structure of optimized networks, new networks could be designed from scratch with a desired feature. Also, we would like to investigate other cost functions that might lead to onion-like structures, and the case of weighted networks, as they are closer to real applications.

**Acknowledgments** This work was supported by the Swiss National Science Foundation under contract 200021 126853, the CNPq, Conselho Nacional de Desenvolvimento Científico e Tecnológico, Brasil, the European Research Council under grant FP7-319968-flowCSS, and ETH Zurich Risk Center.

## References

1. Albert, R., Jeong, H., Barabasi, A.: Error and attack tolerance of complex networks. Nature **406**, 378–382 (2000)
2. Alvarez-Hamelin, I., Dall'Asta, L., Barrat, A., Vespignani, A.: k-core decomposition: a tool for the visualization of large scale networks. Adv. Neur. In. **18**(16) (2005)

3. Buesser, P., Daolio, F., Tomassini, M.: Optimizing the robustness of scale-free networks with simulated annealing. In: Dobnikar, A., Lotrič, U., Šter, B. (eds.) Adaptive and Natural Computing Algorithms, vol. 6594, pp. 167–176. Springer, Berlin (2011)
4. Kirkpatrick, S., Gelatt, C.D., Vecchi, M. P.: Optimization by simulated annealing. Science **220**, 671–680 (1983)
5. Król, D.: Propagation phenomenon in complex networks: theory and practice. New Gener. Comput. **32**(3–4), 187–192 (2014)
6. Latora, V., Marchiori, M.: Efficient Behavior of small-world networks. Phys. Rev. Lett. **87**, 198701 (2001)
7. Newman, M.E.J.: Assortative mixing in networks. Phys. Rev. Lett. **89**, 208701 (2002)
8. Newman, M.E.J.: Networks: An Introduction. Oxford University Press, Oxford (2010)
9. Schneider, C.M., Moreira, A.A., Andrade, J.S., Havlin, S., Herrmann, H.J.: Mitigation of malicious attacks on networks. Proc. Nat. Acad. Sci. USA **108**, 3838–3841 (2011)
10. Wu, Z.X., Holme, P.: Onion structure and network robustness. Phys. Rev. E. **84**, 026106 (2011)

# Chapter 10
# Cancer—A Story on Fault Propagation in Gene-Cellular Networks

**Damian Borys, Roman Jaksik, Michał Krześlak, Jarosław Śmieja and Andrzej Świerniak**

**Abstract** We discuss problems related to propagation phenomena in biological networks. As an example we consider processes leading to carcinogenesis and development of cancer, seen as a complex genetic disease from a system theoretic point of view. We present particular regulatory mechanisms which make the cell cycle a fault tolerant system. Then we indicate weak points in this system leading to mutagenesis and cancer progression. The next stage in this cascade of events is related to an angiogenic switch, which in turn may be treated as a trigger of metastasis. All these processes result from communication, competition and subordination between normal and cancer cells. We illustrate interaction processes by models based on evolutionary games and spatial evolutionary games, which describe propagation phenomena in time and space.

## 10.1 Introduction

Biological networks belong to the most complex real world networks (see [61]), in which propagation phenomena are still far from being completely recognized. We present an example of such complex processes, cascade of which leads to carcinogenesis and development of cancer disease. More precisely, we treat cancer as a result of

D. Borys · R. Jaksik · M. Krześlak · J. Śmieja · A. Świerniak (✉)
Systems Engineering Group, Faculty of Automatic Control, Electronics and Informatics,
Silesian University of Technology, Akademicka 16, 44-100 Gliwice, Poland
e-mail: damian.borys@polsl.pl

R. Jaksik
e-mail: roman.jaksik@polsl.pl

M. Krześlak
e-mail: krzeslak.michal@gmail.com

J. Śmieja
e-mail: jaroslaw.smieja@polsl.pl

A. Świerniak
e-mail: andrzej.swierniak@polsl.pl

D. Król et al. (eds.), *Propagation Phenomena in Real World Networks*,
Intelligent Systems Reference Library 85, DOI 10.1007/978-3-319-15916-4_10

fault propagation in biological networks built of signaling pathways in gene-cellular system.

Cancer is a disease of genes and molecules, and our increasing understanding of the processes related to its genesis makes it possible to develop exciting new strategies for prevention, avoidance and correction of changes leading to carcinogenesis and progression of the disease (see [17]). An explosion of interest in research concerning these issues is related to development of the so called systems biology [102], which brought advances in our knowledge on deregulation of genomes and alteration in metabolic and signaling pathways leading to cancer cells phenotype and contributed to search for molecular targets for anticancer strategies.

The starting point of the disease remains still one of the most enigmatic and exclusive aspects of cancer pathogenesis. How does it happen that in an almost perfect fault tolerant control system such as cell cycle some cells may change their genotype through mutation and thus start the process of carcinogenesis? Discrete events contributing to this process are followed by propagation of faults in the regulatory network that controls intracellular processes through the system of signaling and regulatory pathways. The cells may migrate as pathfinders or pathgenerators (see [106]), as competing individuals or cooperatively. Discovery of processes determining interaction between different cells, their communication, competition and subordination is still a challenge for researchers. These difficulties lead to development of various approaches to modeling and analysis of the regulatory networks. Cancer cells are subject to a variety of stress factors which provide selective pressure acting on increased variation created by progressive deregulation of cancer cells genomes. Tumors, like normal tissues, have physiological constraints on growth such as oxygen and nutrients availability. For this reason, tumors remain in dormant state unless they develop in a well vascularized area or are able to recruit their own vasculature. Tumors do this by the so called angiogenic switch, a discrete event in tumor development that can occur at different stages in tumor-progression pathway. This event becomes, in turn, the beginning of metastasis responsible for about 90 % of deaths from cancer. The important finding is that it is necessary to take into account both time and spatial distribution of signal transduction among cells in studies related to carcinogenesis, tumor growth and development, its motility and invasion (see [67]).

Our contribution is based on system engineering rather than systems biology way of thinking, although the border between them is fuzzy. In the following section we present our understanding of mechanisms behind properties of fault tolerant control systems responsible for proliferation of eukaryotic cells. They constitute the first, intracellular network under consideration, in which transmitted signals allow switching into consecutive phases. We concentrate on several known feedback regulators and monitoring systems that control the cell cycle. The subsequent section is devoted to failures which may evade this control machinery and propagate in the biological network leading to the development of cancer. Section 10.4 deals with a subsequent step in this cascade initiated by an angiogenic switch, which is responsible for progression of the disease. Finally, Sect. 10.5 describes one of the approaches that may be used to model both time and spatial phenomena related to cell to cell interactions in the discussed propagation processes. It is based on the theory of evolutionary games and spatial evolutionary games, and we discuss various issues related to this approach illustrating them with our original results obtained for a four-phenotype model of extracellular signal interactions. This approach enables at least a quantitative analysis of phenomena addressed in Sects. 10.2–10.4.

## 10.2 Cell Cycle as a Fault Tolerant Control System

### *10.2.1 Cells—The Building Blocks of Life*

Every living cell may be considered as a device capable of executing various chemical processes necessary for the organism to survive, develop and reproduce. The basic elements of animal cells include cytoplasm and its organelles which function as factories where various chemical processes take place, and cell nucleus, which stores the genetic information used to control all of the cell functions. The most important element of cell nucleus is the deoxyribonucleic acid (DNA) which can be considered as a set of instructions on how to build individual elements of the cellular machinery and how to control their functions.

The instructions are encoded based on a specific order of four DNA subunits, called nucleotides, used to organize information, control its availability and allow to copy its specific elements termed genes. Genes can be considered as regions of the DNA which contain information on the structure of specific proteins encoded using a universal genetic code, conserved trough all living organisms. In a process called transcription the information contained in genes is copied to a matrix build from ribonucleic acid (RNA). Each matrix is then used to produce multiple copies of a specific protein in a process called translation. The most important feature of this mechanism is that its efficiency depends not only on the structure of the produced protein but also on the concentration and activity of various other molecules involved in its production [2].

Proteins have various functions in the cell and in many cases proteins of the same type are used in different processes depending on cell conditions and various protein modifications. Proteins are the building blocks of the cell, maintaining its shape and internal structure [2]. They also catalyze biochemical reactions [75], carry molecules from one place to another [83], control the efficiency of gene expression [95] and bind to other molecules, thereby controlling their stability [42]. Proteins variability and versatility combined with their changing concentration creates a system of great complexity, which allows the cell to initiate various processes based on its current requirements. To make this possible, the cell requires a sophisticated control system that can respond to various external signals and its internal "sensor" readings.

### *10.2.2 Information Processing and Signal Propagation*

The mechanism of intracellular signal propagation is mainly based on the balance between production and degradation rates of various molecules. The molecules can

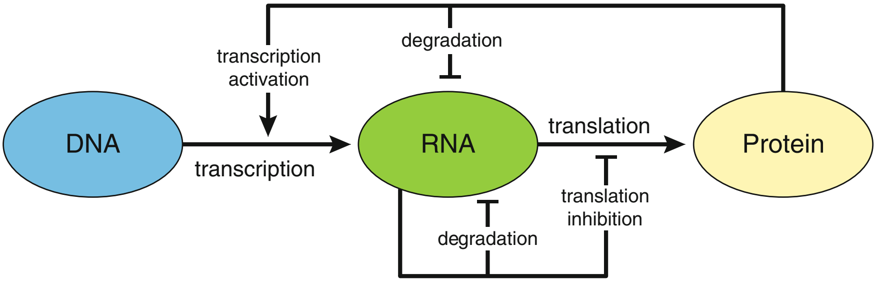


**Fig. 10.1** The basic mechanisms of intracellular protein level regulation

interact with each other, thereby affecting their stability or promoting the production process. The basic regulatory mechanisms of such type include interactions of specific proteins (transcription factors) with the DNA, thus initiating the RNA production process, and protein-RNA interactions or interactions between various RNA types that usually lead to an increased RNA decay (Fig. 10.1).

The regulatory mechanisms, despite utilizing different kinds of molecules, are all based on a target recognition processes of high specificity, which allow binding of just one or few of molecules out of many thousands encountered. The selective binding depends on the formation of many weak non-covalent bonds, like hydrogen bonds, ionic bonds, and van der Waals attractions, which allow to create a stable connection between the molecules if the affinity between them is high or a short-lived interaction if the affinity is weak. Protein modifications form another element of intracellular signal propagation systems. Almost every protein in the cell can be chemically modified after its synthesis, affecting its activity, stability and cellular location. Such modification may initiate or inhibit certain processes including those responsible for cell metabolism, differentiation or immunological response [58].

Two of the most common modification mechanisms utilized by signal propagation systems include protein phosphorylation and ubiquitination. Phosphorylation, and the reverse process of dephosphorylation, work by adding or removing an additional phosphatase group to specific protein residues. The enzymes which take part in these processes (kinases and phosphatases) target proteins from all classes, like other enzymes, structural proteins, or various signaling molecules. Such mechanism creates a switch that can turn on or off specific functions of various proteins by changing their activity or affecting molecule binding capabilities [27].

### *10.2.3 Intracellular Regulatory Systems*

There are over 26,000 genes in a human cell which encode about 47,600 distinct RNA templates [79]. This can lead to the creation of over 120,000 unique proteins [21]. Incorporation of feasible protein modifications further extends the regulatory capabilities of the cell, creating an incredibly sophisticated system. In order to maintain its stability, the cell utilizes thousands of both positive and negative feedback loops responsible for the control of various processes.

Negative feedback loops are an essential component of complex organisms allowing them to maintain homeostasis by regulating synthesis and turnover rate of various chemical substances. Positive feedback loops are not that common but still they play an important role in the intracellular regulation. Positive feedback loops work by either activating protein A through protein B which is responsible for protein A activation or more commonly through double negation where A blocks an A-inhibiting factor B [78]. Such mechanisms are used when the regulatory system is required to take immediate action in response to, for example, detection of DNA damages [59, 80]. Rapid regulatory system response is a key component of many signal transduction pathways and plays an important role in the cell cycle associated processes that allow the cells to grow and divide, making it one of the most sophisticated regulatory systems.

### *10.2.4 The Cell Cycle Clock*

Cell cycle is a very complex sequential process controlled by a variety of mechanisms which make up a system similar to a clock governing cellular proliferation. The entire process is controlled by specific checkpoints which stop the cycle unless a specific signal is received, ensuring that all stages are properly timed and proceed in an appropriate order. Moreover, no transition from one stage to another takes place unless certain criteria are met.

The typical eukaryotic cell cycle lasts around 24 h and includes four phases $G_1$, $S$, $G_2$ and $M$. $G_1$ phase, also called the growth phase, is used by the cell to increase its size and prepare proteins necessary for the synthesis phase ($S$), where the DNA is replicated creating two identical copies of each chromosome. In the $G_2$ phase the cell growth continues, preparing it for the mitosis phase ($M$), in which daughter chromosomes are separated and the cell divides. The duration of each phase varies significantly between different cell types, although in typical cells the $G_1$ and $S$ phases are the longest ones while $M$ phase is the shortest (around 5 % of the entire cycle time). Relative duration of each phase for a typical human cell is presented on Fig. 10.2.

The time dependency of the cell cycle is maintained by a set of proteins termed cyclin-dependent kinases (Cdks) and their association with regulatory subunits cyclins that are responsible for Cdks activity. Cyclins represent a group of proteins which, as the name implies, undergo a cycle of production and degradation through the entire process. Cyclins bind to the Cdks, which usually have a constant level, but require cyclins to gain their protein kinase activity. Because of that, cyclin-Cdk activity also oscillates and the phase of the oscillations determines the moment of initiation of specific processes and allows to trigger an additional control

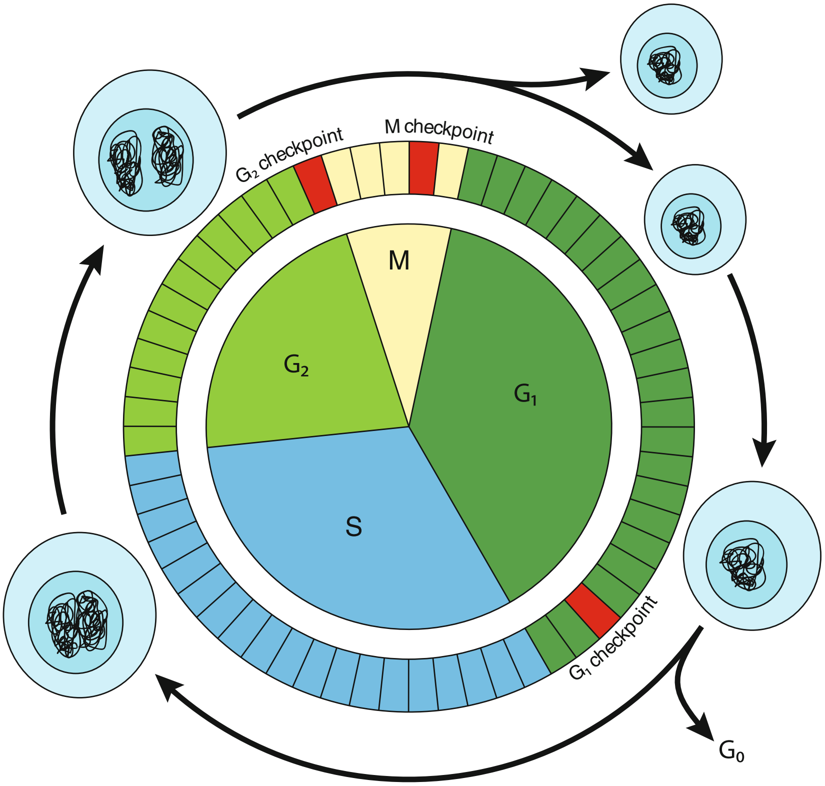


**Fig. 10.2** Cell cycle phases with their relative times matching a typical fast proliferatinghuman cell

processes if the level of specific cyclin is high or low for an abnormally extended amountof time.

Initiation of specific cell cycle events is dependent on the oscillations in the cyclin-Cdk complexes, for example, activation of M-phase cyclin-Cdk complex triggers mitosis, while the initiation of *S*-phase requires high amount of active *S*-phase cyclin-Cdk complexes. Activity of such complexes depends on various regulatory mechanisms, similar to those used in other crucial intracellular processes, like binding of inhibitory proteins, protein modifications (phosphorylation, ubiquitination), and control of the transcription rate for genes coding for various elements of the cell cycle-related pathways.

The fluctuation in activity of cyclin-Cdk complexes leads to changes in the phosphorylation level of various proteins that either activate or inhibit specific elements of the cell cycle, like mitosis or DNA replication. Cyclins also help to guide Cdks to specific target proteins providing very high specificity of the cyclin-Cdk complex dependent regulation, in which only a certain substrate protein or small group of proteins is affected.

### 10.2.5 Cell Cycle Checkpoints

The cell cycle checkpoints are one of the most important elements of the entire process, providing correct order of all events. The the next step can be initiated only if the previous one was completed successfully and if the cell state, which sometimes changes rapidly due to environmental conditions and extracellular signals, is suitable to carry on to the next step.

There are three main checkpoints in the cell cycle at the end of $G_1$, $G_2$ and $M$-phases (Fig. 10.2). The main role of $G_1$ checkpoint is to prevent replication of damaged DNA. It does that by monitoring the level of DNA damage whose presence activates a specific protein kinase—ATM [59]. ATM is responsible for phosphorylation of Cdks2 family of proteins which activate the cell cycle arrest mechanism, providing time necessary to initiate the DNA repair processes [77]. If the repair is impossible or inefficient, the cell will arrest the cycle until certain criteria are meet. This prevents various problems that might occur if the cell progressed prematurely to the next step and replicate a damaged fragment of the DNA [14]. $G_2$ checkpoint also monitors the DNA damage level and additionally ensures that the DNA was properly replicated before the cycle moves to the mitosis phase. By inhibiting B1/Cdc2 cyclin the cell reduces the level of Cdc2 protein which prevents it from entering the $M$-phase of the cycle until the replication is complete and/or the DNA damages are repaired [14]. $M$-phase checkpoint is the final security mechanism that controls the genome integrity, ensuring that each daughter cell receives its complete set of newly replicated chromosomes [22].

In general, the checkpoints prevent a catastrophic cell division, when daughter cells receive only a part of the DNA, or the DNA received is damaged, which might lead to genomic instability. The control system which provides that is highly responsive to information received from each of the controlled stages, changing the course of each process when necessary. The regulation usually operates through process inhibition instead of stimulation, leading to the cell cycle arrest. The signals are triggered if at least one of the required conditions is not met, like appropriate level of nutrients necessary to conduct the process, sufficient growth level or completed duplication of various cellular components [2]. Such design has a significant advantage that can be justified from the control theory point of view. A system is much more fault resistant if it is required to detect a single "stop" signal generated by at least one of its elements rather than detect an appropriate level of "go" signal which would indicate that all preceding processes completed successfully. Utilizing these checkpoints provided the cells with extreme robustness in the course of the evolution. However the performance of checkpoint-based control mechanisms would be insufficient without the p53 protein signaling pathway, considered as the guardian of the genome integrity.

Since the control system regulating the cell cycle is so effective, mathematical models describing its dynamics often do not entail its details. Instead, it is usually based on the compartmental approach to modeling (in which compartments represent subpopulations being in the same phase of the cell cycle). This is particularly useful when the goal of modeling is to predict responses not of an individual cell, but of a cell population, to external control actions representing, for example, phase-specific therapeutic agents (see [91]).

### *10.2.6 Stress Resistance and Damage Recovery*

The cell cycle is designed to withstand damage that might occur to the cell during one of the cycle phases. This prevents the cell from undergoing an improper division and from passing damaged genetic material to the daughter cells, that would have catastrophic consequences for the entire organism. The most common sources of damage that occur during the cell cycle are genotoxic substances and factors like reactive oxygen species [52] or ultraviolet [86] and ionizing radiation [45], which induce various kinds of the DNA damage. The detection of such damages is not as simple as verifying if the DNA helicase is broken, since apart from generating DNA breaks the factors mentioned above can lead to chemical modifications of certain DNA monomers and single-nucleotide substitutions [14]. The total length of the DNA, which in human cells exceeds three billion nucleotides, requires an extremely sensitive and effective regulatory mechanism that could detect damages and prevent them from being propagated to the daughter cells during mitosis. The regulatory module of the p53 protein provides such mechanism, which when activated by the presence of DNA damage, controls a variety of processes that protect the cell and activate its self-degradation if necessary [80].

p53 is a transcription factor activated by the ATM protein kinase in response to the DNA damages. The main protective role of the p53 protein is to initiate cell cycle arrest mechanisms in either G1 or G2 phase and activate the DNA repair machinery [109]. In normal conditions the level of active p53 is very low, maintained by two feedback loops, one positive and one negative [50], illustrated on Fig. 10.3. The negative feedback involves the Mdm2 protein, which is activated by p53 and at the same time responsible for p53 degradation [51]. The positive feedback, which involves the AKT protein, works through double negation. AKT enhances the Mdm2 mediated p53 degradation, and by itself is negatively regulated by the p53. This allows

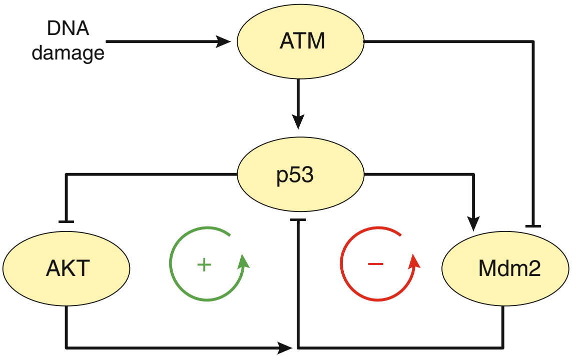


**Fig. 10.3** Simplified model of the p53 protein regulation

p53 to inhibit its own inhibiting factor—AKT, creating a positive feedback loop that significantly increases p53 concentration [18].

In stress conditions ATM disrupts the Mdm2-mediated p53 degradation, leading to rapid p53 elevation [72]. This mechanism works like a countdown clock providing the cell with a certain amount of time to repair the damaged DNA. If the repair is not efficient, the level of active p53 keeps rising, triggering the transcription of proapoptotic genes that initiate the programmed cell death termed apoptosis. The p53 regulatory module ensures that the cell repairs the damages before replicating its DNA. However if the damages are too severe the cell is destroyed in order to prevent damage from propagating to daughter cells, and in turn, affecting the entire organism.

Cell cycle is an extremely well designed regulatory system capable of controlling a complex cell proliferation process even in very harmful cellular conditions created by genotoxic factors. Its robustness and the ability to deal with distinct types of cell damage by the use of various checkpoints and repair mechanisms makes it an almost perfect fault tolerant system.

## 10.3 Carcinogenesis—Failure of the Cell Cycle Control System

As indicated in the preceding section, multiple regulatory mechanisms help to maintain the cell cycle and, ultimately, lead to cell division, after which two cells with the same genetic information begin their life cycles. While there is a lot of redundancies in these mechanisms, when several of their components fail, the errors add up, and instead of preventing individual cells from exerting harmful effects on the organism, they might get amplified. First, these errors propagate through the intracellular signaling networks and, subsequently, they spread in cell populations through the transmission network that allow cells to communicate with each other, for example releasing mitogens that stimulate neighboring cells into entering the division process. Thus, one can view the cross-linked extra- and intracellular processes as a network within the network.

One might distinguish two basic types of control mechanisms that should be taken into account when looking at the cell cycle as an almost failure-resistant system. The first one concerns processes activated when DNA is damaged, as only complete, correct DNA should be replicated (Fig. 10.4). The other type includes the processes that determine the length of cell cycle and are based on the checkpoints mentioned in the preceding section.

### *10.3.1 Failure of DNA Repair Mechanisms*

DNA damage arise naturally in all cells (e.g. 55,000 single strand breaks per cell per day were reported in [60]). Different types of damage are detected by appropriate

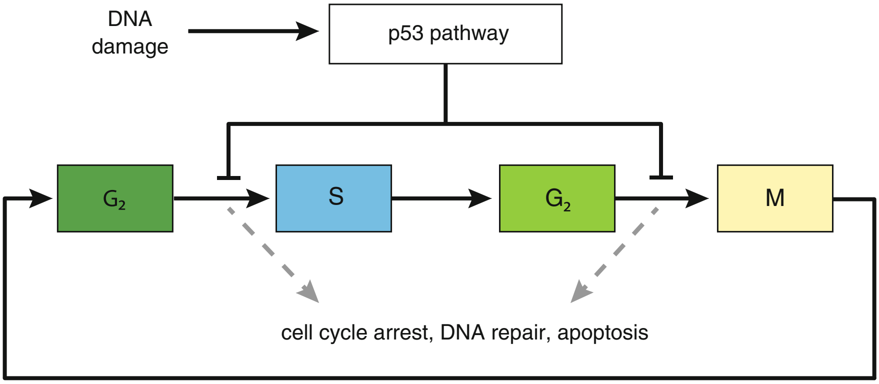


**Fig. 10.4** Two basic mechanisms, destabilizing cell cycle: promoting entry into S phase and blocking the apoptosis pathway

sensory proteins, activating repair mechanisms. If the repair is not possible, which is signaled by a prolonged high level of respective proteins, the programmed cell death, apoptosis, should be initiated. Despite all these mechanisms, mutations, rearrangements and other disruption of genetic information may take place, particularly during DNA replication. They are the most dangerous when they occur in genes that code for proteins controlling either cell cycle or DNA repair processes, since in these cases the error signal propagates through the cell gene regulatory network an later may spread in population, leading to development of cancer. In the case of genes involved in repair processes, the result is loss of genomic stability. For example, mutations in genes coding proteins responsible for initiating double strand breaks recognition and triggering DNA damage response are associated with a high incidence of leukemia and lymphoma [41, 84]. Another example is the chromosomal translocation that leads to a proliferation signal resulting in development of chronic myeloid lymphoma [19]. Mutations in mismatch repair genes are involved in development of diffuse large B cell lymphomas [25]. Large fraction of colorectal tumors show an abnormal shortening or lengthening of dinucleotide repeat sequences, known as microsatellite instability. It arises when mismatch repair is defective due to mutations in mismatch repair genes. Familial forms of breast, ovarian and pancreatic cancer are associated with mutations in recombination modifying genes [65].

The accumulation of DNA defects leads to the increased number of DNA mutations that develop into carcinogenesis. This should be counteracted by, among others, control actions exerted by the tumor suppressor p53. In addition to its involvement in DNA repair, responses to heat shock, as well as other factors, it also plays a critical role in arresting the cell cycle when DNA damage is detected and plays a key role in promoting apoptosis, when damaged DNA cannot be repaired. However, proper functioning of p53 pathways depends on its proper form. In the majority of human cancers p53 is dysfunctional, most frequently due to a point mutation within its DNA binding domain [88]. Such mutation is associated with inducing expression of cell cycle-promoting genes, such as cyclin A, cyclin B1, cdk1, and cdc25C, and to an increase in DNA synthesis [16]. Moreover, mutant p53 enhances proliferation by cooperation with various cell cycle-related factors such as ETS, E2F, and MYC [15]. All this evidence is behind continuous extensive investigation of p53 pathways [49] and development of mathematical models that support experimental research. Depending on their main goal, these models can be either very simple, focused on a single regulatory action [31, 80] or larger, comprising more feedback loops, both positive and negative [55, 56].

One of the mechanisms that supports p53 actions is binding of the p19 (ARF) protein to Mdm2. Since Mdm2 normally promotes p53 degradation (see Fig. 10.3), activation of p19 leads to increasing levels of p53, resulting in cell cycle arrest or apoptosis. Loss or mutation in the p19 gene disrupts this process and may lead to cancer [43, 100]. Another gene, whose transcription is induced by p53 following DNA damage is p21. Since p21 is involved in controlling entry into the S phase, its role in disruption of the cell cycle is described in the subsequent section.

### *10.3.2 Cell Cycle Disruption*

When the mutated gene is associated with cell cycle control, it usually leads to accelerated cell division and failure to respect checkpoints in passing from one phase to another. This property is a characteristic feature of cancer.

It seems that the most critical, with respect to possible alterations of the cell cycle, is the G1-S transition (Fig. 10.5). Though this passage is controlled by many pathways, at least partly redundant, this is the key in triggering error propagation in cell cycle. One of the possible sources of failure has already been mentioned in the closing paragraph of the preceding section. It involves the p21 protein that binds to the G1/S-Cdk and S-Cdk complexes inhibiting their activities and thus blocking entry into the S phase. Since p53 is a transcription factor for p21, if it does not work properly, p21 cannot be induced, which leads to initiation of damaged DNA replication. However, there is a growing evidence that p21 can function as both a tumor suppressor and an oncogene [104].

Replication of damaged DNA is not the only problem that may arise in cell cycle and contribute to the development of cancer. Another one lies in acceleration of the cell cycle, caused by earlier than necessary entry into the S phase (Fig. 10.5). This entry is dependent on increased activity of the E2F protein. It can be promoted either by inducing transcription of E2F gene or by inactivation of the retinoblastoma protein (Rb) that acts as a brake on the cell cycle progression. This, in turn, can be achieved through the activation of the G1-Cdk cyclin (cyclin D-Cdk4), preceded by increased cyclin D1 production.

The cyclin D1 is frequently overexpressed in a wide range of cancers, sometimes coincident with gene amplification or somatic mutations of the gene coding it. A frequent alternative splicing leads to production of cyclin D1b protein that lacks a specific phosphorylation site required for nuclear export, leading to its accumulation in the nucleus and increased interaction with Rb, and, subsequently, promoting entry

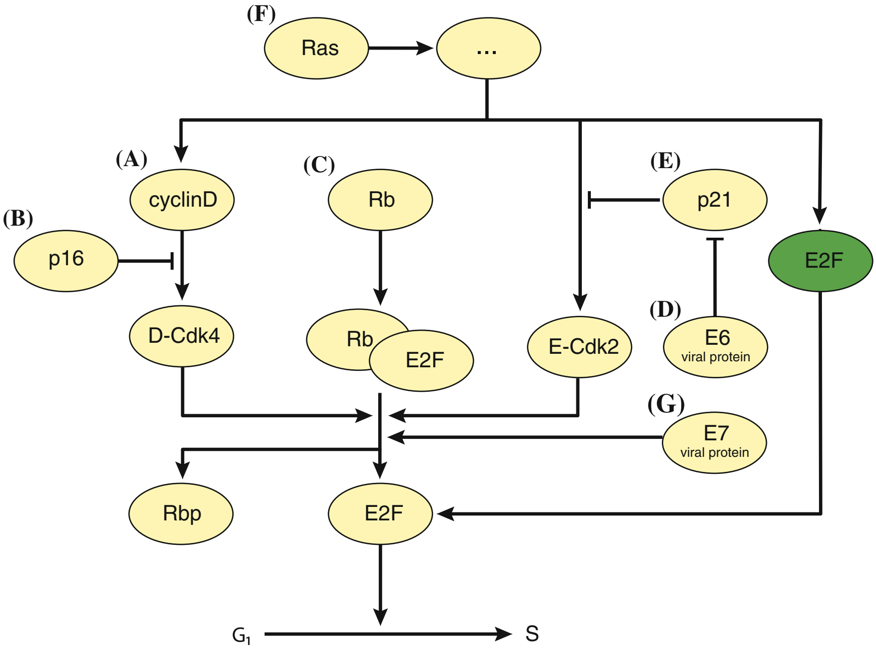


**Fig. 10.5** Interaction network regulating entry into the S phase and sources of its failure in cancer cells: **a** D1 overexpression, **b** p16 inactivation, **c** Rb inactivation, **d** p21 and **e** p27 failures

into the S phase [39]. While this could be prevented by another control mechanism, based on the p16 protein that blocks the formation of an active D1-cdk4 complex, many cancer cells have either a deleted, inactivated or silenced p16 gene [2, 33]. Moreover, some mutations in the p16 gene promote cancer metastasis [20]. On top of this, in some cancers p16 is overexpressed and despite that, these cancers may have poor prognoses [108]. This suggests that our knowledge of even this, relatively small part of the signaling network, is far from complete and caution is recommended, concerning conclusions drawn from experimental work and mathematical modeling that supports it. Another important regulator of the cell cycle is the p27 protein. Its increased degradation, natural in a normal cell cycle, leads to G1/S-Cdk activation, thus promoting entry into the S phase. It has been found that in some tumors p27 is mutated in a way that reduces its stability [71]. The ultimate result of such mutation is, once again, uncontrolled entry into the S phase and acceleration of the cell cycle.

One of the most extensively studied family of proteins is the NF-κB (nuclear factor κB) transcription factor family. It is a key element in many regulatory networks, playing a crucial role in pathways activated by a wide variety of stimuli and environmental challenges. It is also involved in regulation of the cell cycle [63], controlling, among others, different cyclins [73]. Its actions promote cell proliferation and cell growth, as well as block induction of apoptosis by inducing transcription of genes coding inhibitor of apoptosis proteins [32].

One should also remember that environment also contributes to stability of the cell cycle. Extracellular signals from neighboring cells, called mitogens may overcome intracellular mechanisms that block or slow the cell cycle [2]. They act through signaling pathways involving a small GTPase Ras. A mutation in Ras-coding gene may cause it to be permanently active, thus leading to continuous progress of the cell cycle [26, 54, 89, 94] as well as helping to fuel metabolic pathways, supporting growth and division [105]. These mutations are found in about 25 % of human cancers and are highly prevalent in hematopoietic malignancies [103]. On the other hand, viral infections may also lead to changes that promote activation of transition to the next phases of the cell cycle under wrong conditions, and, ultimately, result in carcinogenesis. For example, human papilloma virus produces oncoproteins E6 and E7, which disrupt, otherwise well-performing, regulatory network. The first of these blocks p-53 mediated activation of the p21 protein, while the other inactivates Rb, thus activating E2F and inducing cell cycle progression independent of the G1-S checkpoint Cdks (Fig. 10.5) [1, 82].

The mechanisms described above are only a sample of what experimental research has discovered in recent years. It is clear that the sheer complexity of the regulatory networks under consideration makes the planning of experiments and analysis of their results extremely difficult. This is exactly the point, at which mathematical modeling may prove useful and the reason for a rising popularity of models of large signaling pathways on one hand, and basic regulatory modules on the other. These models help to find missing elements of the signaling network [87], identify kinetic parameters of the processes [37] that can be subsequently used to better experiment planning [36], analyze the effects of external stimulation of these pathways [15, 81]. Among the models describing various pathways and regulatory networks one can easily find those focused on the cell cycle and the mechanisms described in this section (see [7, 24, 99]), allowing to predict the propagation of faults, mentioned above, and its consequences for the cell fate.

## 10.4 Tumor Angiogenesis and its Role in Disease Progression

Tumor—as a result of fault propagation occurring in the cell cycle, has one general feature—fast and uncontrolled proliferation of its cells leading to its unstable growth and development. However, in-situ tumor is able to increase its size only up to some limits, about 1–2 mm in diameter, above which tumor experience hypoxia and acidosis due to the inadequacy of nutrient supply and metabolic waste clearance by vessels. By exceeding this limit tumor starts a new malignant phase, which is often followed by the process of metastasis. Metastasis is the spread of cancer cells into lymphatic and blood vessels, circulate through the bloodstream, and then invade and grow in normal tissues elsewhere. This state forms a direct threat to the body of a host. In other words, a fault emerging during the cell cycle, propagated by replication of mutants, characterized by excessive cell proliferation and their invasion to the neighboring or distant organs, affects the activity of very complex mechanisms at the tissue, organ or whole body level. This fault propagation, surpassing cellular level, can destroy an organism built with billions of cells, if not intercepted or stopped in time.

One of the crucial steps and also necessary conditions for cancer invasiveness and motility is creation and development of an autonomous blood vessel network. During this process, called angiogenesis, new blood vessels are built basing on the existing network of vessels and penetrate the neighborhood of cancerous cells, supplying them with nutrients and oxygen and removing waste products. This is not the only factor determining malignancy of tumor. In the literature [47, 48] one can find six general, distinctive and complementary capabilities—hallmarks of cancer—that enable tumor growth and metastasis. They include: sustaining proliferative signaling, evading growth suppressors (insensitivity to anti-growth signals), resisting cell death (evading apoptosis), activating invasion and metastasis, enabling replicative immortality, inducing angiogenesis. First three of them have been addressed in the preceding section. This section focuses on angiogenesis.

The existence of well-developed vascular network is crucial for normal tissue homeostasis. The process of angiogenesis occurs in healthy tissues at some stage of their development and is one of the factors sustaining their natural growth, also allowing embryogenesis, organs growth etc. During embryogenesis, the development of the vasculature involves the birth of new endothelial cells (a thin layer of endothelial cells lines the interior surface of blood vessels and lymphatic vessels) and their assembly into tubes (vasculogenesis) and sprouting (angiogenesis) of new vessels from existing ones [76]. This process is also important for wound healing, tissue repair or female reproductive cycle. Therefore, in controlled and regulated situations, angiogenesis is a highly indispensable mechanism resulting, in normal conditions, in generally quiescent vasculature. The vascular endothelial cells that form walls of blood vessels rarely divide but angiogenesis can stimulate them to do this.

Tumor cells, like any other cells, depend on nutrients and oxygen supplies. They also need to secrete products of metabolic processes—toxic wastes and carbon dioxide. That is why, taking into account the excessive proliferation rates, without neovasculature the size of tumor volume cannot exceed some limits determined by penetration range of supplies from existing vasculature. Angiogenesis is initiated by cancerous tumor cells producing and releasing molecules that constitute signals to surrounding normal tissue. This signaling activates certain genes in the host tissue, which code for proteins that promote growth of new blood vessels.

### 10.4.1 Pro and Anti-Angiogenic Factors and Mechanism of Angiogenesis

Angiogenesis is regulated by both activator (proangiogenic factor) and inhibitor (antiangiogenic factor) molecules. Normally, inhibitors dominate, blocking growth, maintaining the state of equilibrium in the vascular system. When the need for new blood vessels arises, activator production rate increases and the number of inhibitors decreases. This prompts the growth and division of vascular endothelial cells and the formation of new blood vessels begins (Fig. 10.6). Thus during tumor progression,

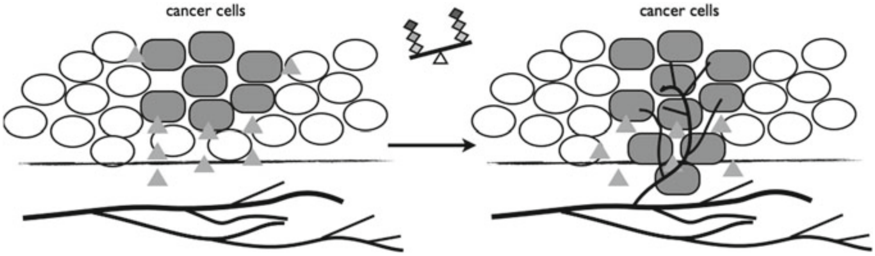


**Fig. 10.6** Schematic presentation of tumor angiogenesis and its role in tumor growth

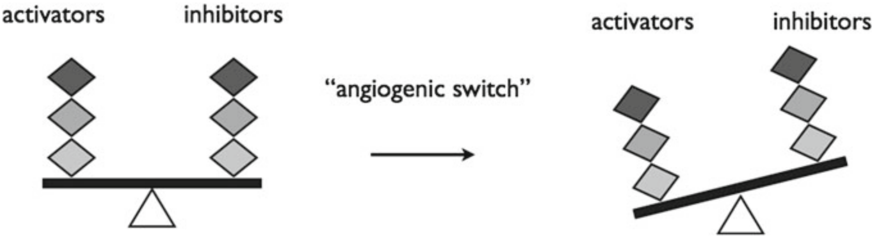


**Fig. 10.7** The angiogenic switch

an "angiogenic switch" is activated causing normally quiescent vasculature to sprout new vessels that help sustain expanding neoplastic growths [6, 46]. This angiogenic switch is considered as a discrete event in the tumor development involving a change in the local equilibrium between activators and inhibitors of angiogenesis and is a result of a tilt towards pro-angiogenic regulators (Fig. 10.7). This signaling activates particular genes in the surrounding tissue that produce proteins which promote the growth and sprouting of blood vessels and thus tumor development. First studies on this phenomenon date back to the sixties and early seventies of the last century, when the group of Folkman [35] described the hypothesis that tumors produce diffusible factors that evoke angiogenesis. They also formulated a suggestion that identification of key molecular players driving tumor angiogenesis could result in effective strategies to inhibit it, and ultimately enable tumor starvation.

There are several mechanisms used by cancer cells to launch new vessels formation: sprouting from existing vessels (angiogenesis), recruitment of bone marrow-derived endothelial progenitor cells to form new vessels (vasculogenesis) and splitting a single vessel into two (splitting angiogenesis). Moreover cancer cells can intercept existing blood vessels or incorporate itself into the vessel wall to obtain nutrients for their growth.

The process of tumor angiogenesis occurs without superordinate control and the resulting vascular network is structurally abnormal. Anatomically, tumor vessels are dilated, tortuous, and saccular with disorderly interconnection and branching [44]. Also, unlike the normal tissue vasculature, sites of increased as well as reduced vessel density appears. Angiogenesis in normal tissues, for example, during wound healing, is strictly controlled leading to a regular network. It justifies the description of tumor as “a wound that does not heal” [30]. Tumor abnormal vasculature alters also tumor microenvironment and allows for growth and progression of tumor. Despite numerous efforts to explain the process, it is still not clear why cancer so easily breaks down the regulatory pathways involved in control of angiogenesis.

More than a dozen different proteins and molecule types have been identified as proangiogenic. Among them, two proteins are reported to be the most important for tumor growth: the vascular endothelial growth factor A (VEGF-A, also known as VEGF) and the basic fibroblast growth factor (bFGF). Both are produced inside tumor cells and then secreted into the surrounding tissue. When they reach endothelial cells (EC), they bind to specific receptors on the cells membranes. This activates an internal signaling cascade, that finally promotes expression of specific genes in nucleus of the endothelial cells, which are responsible to make products needed for new endothelial cell growth, and starts further steps toward the creation of new blood vessels (Fig. 10.8).

The first of these steps involves production of special enzymes (matrix metalloproteinases—MMPs), which are subsequently released from the endothelial cells into the surrounding tissue. The MMPs break down the extracellular matrix—a support material that fills spaces between cells. This permits the migration and division of these motile cells. When their number reaches a certain threshold, new cells organize into tubes and evolve into a mature network of blood vessels, with the help of an adhesion factor, such as integrin α or β [76]. Signaling cascade, starting from VEGF-VEGFR2 (VEGF receptor), promotes contraction of the EC cytoskeleton and weakens cellular connections ultimately causing EC migration [44]. Moreover, perivascular cells (PVCs, a connecting tissue at the periphery of vessels), pericytes as well as vascular smooth cells, around tumor vessels change their characteristics. Normally, they interact closely with ECs to prevent vessel leakage but during angiogenesis they often become detached, facilitating EC movement.

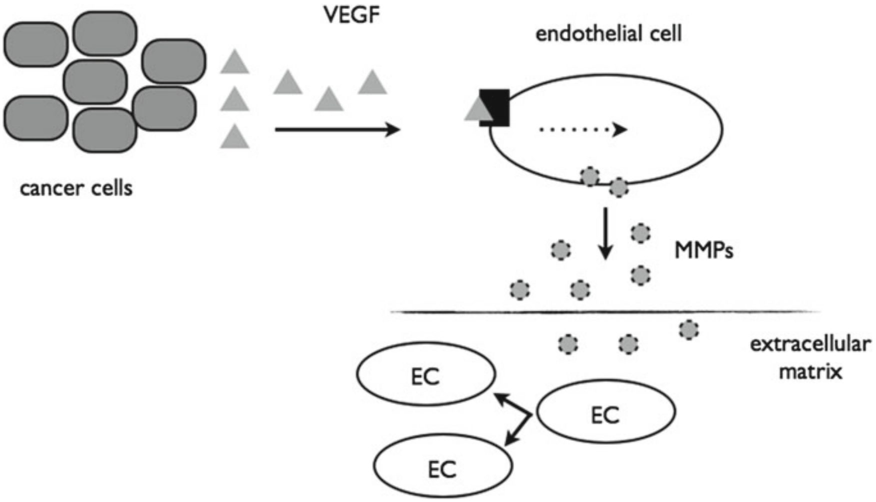


**Fig. 10.8** Signal cascade during tumor agniogenesis

The presence of VEGF and bFGF is not enough to begin blood vessel growth. For angiogenesis to begin, these activator molecules must overcome a variety of angiogenesis inhibitors that normally restrain blood vessel growth. The list of factors can be found in many reviews, e.g. [76, 101]. Among them, proteins called angiostatin, endostatin, and thrombospondin appear to be especially important.

### *10.4.2 Angiogenesis—An Essential Step Towards Tumor Metastasis*

As stated earlier, angiogenesis allows tumor to grow beyond the avascular velocity limit and thus become a malignant type. It is also important that angiogenesis is a critical component of tumor metastasis and that highly vascular tumors have the potential to produce metastases [107]. Developing vascular network into the tumor mass provides an efficient route of exit for cancer cells to leave the primary site and enter the bloodstream. This process facilitates the entry of cancer cells into the blood circulation by building the network of highly permeable blood vessels. Another consequence of tumor angiogenesis, associated with changes at the cellular level—defined by loosely connected ECs and poorly associated PVCs—is leakage of intravascular fluids and plasma proteins. This, together with lack of lymphatic vessels responsible for clearance, in turn, results in increased interstitial fluid pressure (IFP). Together with local vessels collapse, caused by mechanical stress from the proliferating cancer cells, regions of hypoxia (lack of oxygen) and acidosis within tumor appear. Hypoxia is not only responsible for promoting angiogenesis, by promoting production and release of VEGF, inducing HIF1$\alpha$ production (hypoxia—inducible transcription factor), it also activates oncogenes that promote invasive growth and metastasis [12]. As a result, it promotes invasive and malignant behavior of tumor cells. Moreover, tumor cells prove to be much more resistant to hypoxic conditions [44].

Another aspect of a tumor-induced angiogenesis is its impact on development and efficacy of anticancer therapies. Reduced functionality of the immature vascular network decreases the ability of treatment agents to reach their target. There is a lot of ongoing research on anti-angiogenic treatment strategies that would potentially force the tumor into a dormant state. One of the result of studies of tumor angiogenesis is the notion of "normalization" of tumor vasculature [29, 57]. Therefore, nowadays antiangiogenic therapy is considered often to be an essential component of multidrug cancer therapy, especially when combined with chemotherapy (see [90] and references therein). Although tumor eradication in such combined therapy may still be the primary goal, the chaotic structure of the angiogenically created vascular network leads to another target for antiangiogenic agents (the so called pruning effect) to facilitate more efficient drug delivery. The basic idea is to first re-establish vasculature functionality using some initial treatment, and then use a proper killing agent.

The result of unrepaired fault in cell cycle can be a mutant cell whose proliferation is disrupted. Having reached all mentioned hallmarks of cancer this mutant and faulty cell has a potential to grow excessively and evade apoptosis mechanism. By replication in a healthy tissue environment, faulty genetic information penetrate host body.

When some critical mass is reached, this faulty group activates signaling cascade that allows new vessel creation and further growth with possible invasion. From this point of view angiogenesis is one of the crucial steps in multiphase tumor progression and can be promoted by the discrete angiogenic switch, like a light switch.

## 10.5 Are Cancer Cells Good Players?

In 1997 Tomlinson and Bodmer [98] proposed to use methods of Evolutionary Game Theory (EGT) for modeling interactions and communication in a population of cells. This theory combines tools of the game theory with present knowledge of population biology and evolution [68, 85]. EGT differs from the standard game theory by deviating from the assumption about a rational decision making by competing players to treating strategies as phenotypes of individuals, acquired through the evolution. Moreover, in this approach the players represent subpopulations, containing individuals with different phenotypes (strategies), who can cooperate or compete for resources. As a result of different environmental adaptations following a sequence of games through the time (generations), the population may tend to stabilize its structure, at the same time gaining a stable monomorphism or polymorphism of population phenotypes. Such state is called to be evolutionary stable. The evolutionary stable strategy (ESS) is defined as a phenotype that, if adopted by the vast majority of a population, will not be displaced by any other phenotype [69].

Classical models of cancer development assume that mutations which promote development of cancer cells affect only individual cells in which they occur [34, 64, 97]. However, recent studies point out that tumor cells are able to adopt various genetic strategies which may influence the rate of their own development. Moreover, a mutation in one cell can also affect neighboring cells [98]. It may be combined with cooperative behaviors or competition for resources, such as space, oxygen and nutrition that occurs between different subspecies within the same tumor [70]. This leads to a conclusion that tumors should be analyzed as complex ecosystems or networks [3, 23] in which internal communication between tumor cells, and between tumor and normal cells, their competition for resources, hierarchical subordination, and collaboration play an important role in cancer development and differentiation or disease transmission and reaction to stresses including therapy [8, 40]. In other words, the development of the disease and its progress depend on propagation of results of interactions between individual cells in the network both in time and space.

### *10.5.1 Evolutionary Games, Spatial Evolutionary Games and Mixed Spatial Evolutionary Games*

Different cells with different phenotypes participate in evolutionary games. Diverse correlations, interactions and cells coexistence in population have been studied, taking into account the possible domination of tumor cells (phenotypes acquired by mutations). Tomlinson and Bodmer [98] have been followed by others, who considered production of cytotoxic substances, production of growth factors leading to tumor angiogenesis [5], invasion and metastasis [66], tumor-environment interactions [38], the radiation-induced bystander effect [92], resistance to chemotherapy and p53 vaccine [9], interactions between osteoclasts and osteoblasts [28], tumor-stroma interactions [40], interaction between different tumors [9] and other issues (see [8, 93] for a survey and other references). Thus, the basic phenomena described in the previous sections could be modeled by this machinery.

Roughly speaking, the evolutionary game is defined by the pay-off table (matrix) in which both rows and columns represent possible phenotypes and the entries are changes in fitness resulting from interaction between two phenotypes. Evolutionary strategies are defined as fractions of phenotypes in the population.

Using EGT we are able to predict whether the given population has a tendency to become heterogeneous or rather only one phenotype will prevail, dominating the whole population. To track propagation of results of the game in the network in time, we may use replicator dynamics equations [53] describing the fate of population in time starting from initial state and converging to the equilibrium defined by ESS. If $x_i$ $(i = 1, \dots, n)$ denote the evolutionary strategies then their replicator dynamics is defined by the set of $n - 1$ equations:

$$\dot{x_i} = x_i(E(i) - E(x))$$

where $E(i)$ denotes the average profit of the strategy $i$ in the population defined by $x$, and $E(x)$—the average profit of this population.

However, because of assumptions about perfect mixing it gives only mean field results. As a result, it is not possible to take into account effects of local arrangements on intercellular interactions.

The machinery of EGT supported by replicator dynamics enables analysis of time evolution of phenotype structure in cell populations. On the other hand, it gives no information about spatial distribution of these phenotypes in tumors. Incorporation of this information is possible when the methodology of spatial evolutionary games theory (SEGT), which enables study of players' allocation, is applied. Moreover, the use of SEGT enables consideration of propagation of modeled phenomena in the network.

This is why they have become very popular recently, although they are based on the quite old idea of cellular automata [74]. Spatial tools have already been used in modeling of carcinogenesis [4]. The line of reasoning presented there has been the starting point for our analysis, as the most suited to the applications focused on in our investigations The spatial games are played iteratively on a lattice forming torus and each tie in a competition is solved randomly. The following steps are performed every iteration [4]: payoff updating—the sum of local fitness in the neighborhood, removing players—cell mortality, reproduction—defining which phenotype will be on an empty place. There are three ways of including cell mortality in terms of updating the lattice [4]: synchronous—all cells are replaced in accordance to the lattice from previous iteration, asynchronous—one cell is chosen randomly, semi-synchronous—10 % of random players participate in the game.

Semi-synchronous updating is used in analysis presented here. Numerous studies and simulations (e.g. [4, 62, 66, 92]) show that this method reflects a biologically relevant situation. Indeed, synchronous updating would introduce a kind of a global controller of the system (everything is replaced at one time instant), while during asynchronous updating small cell clusters could not be removed. The next step of the algorithm is the replacement for chosen players. The important factor for this phase is the local adaptation, so the sum of eight scores (number of players may be different for different neighborhoods) from cell–cell interaction is calculated, according to a pay-off matrix. Basically, in the examples that follow two kinds of reproduction are presented: deterministic—the winner is the strongest player, probabilistic—local adaptation is divided by sum of local fitness in a neighborhood. According to the authors the latter shall allow phenotypes with lower fitness, but with a better spatial arrangement to dominate the population. Two other methods have been introduced in [62]: quantitative—suggesting correlation between players with the same phenotype (a counterpart for the probabilistic one), and switching—if the differences between adaptations are relatively big, then the quantitative reproduction is the choice (a chance for weaker players). In the appropriate case the deterministic reproduction shall be used. To define a threshold responsible for the choice of one of the reproductions, an additional correction factor, given as a ratio of minimal and maximal fitness has been added.

In the case of application of SEGT to analyze cancer cells behavior, the question that arises is whether each cell has only one strategy (represents one phenotype) or rather it should be treated as containing different strategies. The new idea which we have used in modeling spatial effects associated with evolution of cancer cells is related to their heterogeneity. It leads to the conclusion that cancer cells should be considered as representing different phenotypes at the same time, described by frequency of occurrences. The spatial games resulting from this assumption will be called mixed spatial evolutionary games (MSEG).

Modification of the way spatial games are used requires the change in definition of the local fitness (adaptation). It is defined in a way similar to an expected result of the game with mixed-type strategies. The result given by each pair of strategies is multiplied by their frequency of occurrence. Hence, the analysis is more complex and difficult, due to an increased number of feasible spatial structures. Nevertheless, for simplification, both types of spatial games may be represented in a way similar to the mean-field models.

The new formulation of spatial games also defines mortality of the cells in a different way. Here, chosen player stays alive and either its phenotypes ratio is changed or it affects cells in the neighborhood. Additionally to two basic reproductions (deterministic and probabilistic) at least three additional could be added for the mixed spatial games: weighted mean of the strongest players—the weighted mean accordingly to players payoffs is taken, weighted mean of the best clusters—players are organized into clusters and the weighted mean is calculated for players in the strongest cluster, spreading reproduction—mentioned previously possibility to impact surrounding cells. Players with smaller payoff (multiplied by the correcting factor) are taken into account for the weighted mean. For mean weighted reproductions an additional parameter (factor) is needed to define the number of cells or clusters for the computation. Switching reproduction defined previously for SEGT may be also used for MSEG. In this case switches take place between the deterministic reproduction and the weighted mean of the strongest players).

### *10.5.2 Four-Phenotype Model of Interaction Between Tumor Cells—Time and Space Propagation*

EGT-based theory may give an answer to the question if there exists a stable equilibrium (ESS) between different phenotypes (strategies, clones) leading to a strong heterogeneous structure of cancer cell population. It also provides clues as to how such structure might depend on parameters, which characterize interactions between tumor cells and environment, and the initial distribution of phenotypes in the population (initial conditions). It should be stressed that analysis of evolutionary stable strategies allows to study asymptotic properties of the population only. Unfortunately, in almost all published studies on EGT models the analysis is limited to two or three phenotypes. The exception is our paper [93], in which interactions between four different phenotypes of cells are illustrated using three-dimensional simplexes and time courses. As far as we know, the only other work which considers four phenotypes is [10]. However, instead of studying different equilibrium points between phenotypes and their dynamics, the authors analyzed there only the final results (different subpopulations), with respect to changes of fitness parameters. It is important to notice that the dimension of replicator dynamics equations in the case of three phenotypes is equal to two, which means that complex dynamical behaviors typical for nonlinear dynamics should be absent. In our opinion, it is one of the major disadvantages of the small number of considered strategies. An important finding is that four-phenotype model implies third order dynamics of replication, which enables existence of complex dynamical behaviors, including strange attractors. This may be a crucial hallmark of evolutionary game theory analysis. To illustrate advantages of our approach to analysis of increasing number of strategies let us consider the model which combines two classical models of Tomlinson [96, 98] (Table 10.1). The model contains four different strategies/phenotypes of cells:

*A* production of a growth factor in a paracrine fashion;
*P* production of cytotoxic substance harmful to the neighbors;

**Table 10.1** Proposed pay-off matrix

| Strategies | A | P | Q | R |
|---|---|---|---|---|
| A | $1-i+j$ | $1+j-e+g$ | $1+j-h$ | $1+j$ |
| P | $1-i+j-f$ | $1-f-e+g$ | $1-h$ | $1-f$ |
| Q | $1-i+j$ | $1-e$ | $1-h$ | 1 |
| R | $1-i+j$ | $1-e+g$ | $1-h$ | 1 |

$Q$ resistance to the cytotoxic substance;
$R$ neutrality

Parameters used to define the measure of fitness are given by: the profit from growth factors ($j$), the cost of producing the growth factors ($i$), the harmful effects of cytotoxins ($f$), the cost of producing cytotoxins ($e$), the profit from affecting neighbors by cytotoxin ($g$), the cost of resistance to cytotoxin ($h$). Assuming that the basic cost of the interaction of two individuals is 1, the changes in the measure of fitness defined by the entries of the matrix could be easily deduced. For example, when A meets A, it benefits from production of growth factors but also covers the cost of such production. In what follows, by $E(k)$ we will denote the average profit of the strategy in the column $k$ in population with fractions of phenotypes defined by A, P, Q and R. In the Tomlinson model of the so called angiogenic games the only condition to reach a stable evolutionary state is that the cost of producing growth factors $i$ should be smaller than the benefit $j$. In the extended model the expected pay-offs (the sum of the products of frequency and pay-off) are defined as:

$$\begin{aligned} E(1) &= 1 - i + j - f \cdot P \quad & E(2) &= 1 - e + j \cdot A - f \cdot P + g \cdot (A + P + R) \\ E(3) &= 1 - h + j \cdot A & E(4) &= 1 + j \cdot A - f \cdot P \end{aligned}$$

To achieve quadruple equilibrium following relations should be satisfied:

$$\begin{aligned} &E(1) = E(2) = E(3) = E(4) \quad && E(1) = E(4) \rightarrow A = (j - i)/j \\ &E(3) = E(4) \rightarrow P = h/f && E(2) = E(3) \rightarrow Q = (g - e)/g \\ &\quad R = 1 - A - P - Q \end{aligned}$$

For the polymorphic coexistence between all strategies, each expected frequency has to be constrained to the values between 0 and 1. If the equations above are not satisfied, the results may lead to points that indicate different than quadromorphic populations. The equilibrium point could be either an attractor or a repeller, and the population itself may be unstable. The large number of parameters and four phenotypes cause that the analysis of the model is not as trivial as in the case of separate models. To illustrate the feasibility of the model final states, we can present them in relation to two parameters. Figure 10.9 shows that different monomorphic and polymorphic populations may be obtained for various values of parameters. The disadvantage of this approach is that the dynamics is not shown. Moreover, the exact ratios of phenotypes are arbitrarily assumed and the simulation was performed only for one set of initial frequencies (in the presented case they are all equal). Some basic dependencies may be seen at the first glance, like threshold value of $e$ equal 0.5 which is related to the value of $g$. However, the interpretation of other results is not so obvious, for instance the small area of A, P, Q population just above the quadromorphic one. The second simulation has been performed for different values of $f$ and $i$. The results are even more distinct not only in terms of quantitative results, but also in terms of the shapes of regions. Due to a very large number of different results and combinations of the parameters, we discuss only the case when the population

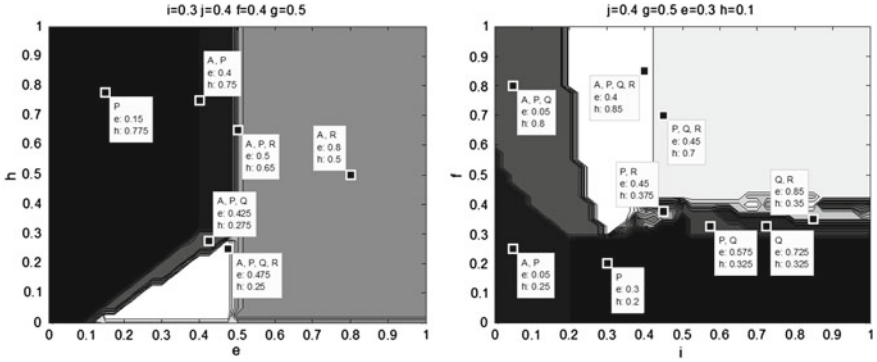


**Fig. 10.9** Different subpopulations in accordance to changing parameters

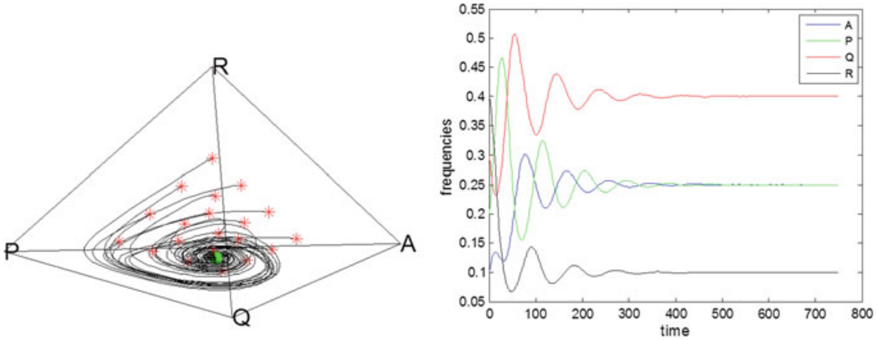


**Fig. 10.10** Mean-field results for i = 0.3, j = 0.4, f = 0.4, g = 0.5, e = 0.3, h = 0.1

is quadromorphic. The EGT analysis (the mean field model) (Fig. 10.10) shows that the steady state is reached in oscillatory way. Q-cells dominate (due to a relatively small *h*), P- and A-cells have got the same value of final frequencies of occurrences and the smallest fraction in the population is constituted by R-cells.

SEGT model simulation (Fig. 10.11) was performed for a lattice generated randomly and only once (different initial lattices and stability of the SEGT would also be an interesting subject of studies). Probabilistic and deterministic reproductions provided results more or less similar to the mean-field model. In most simulation iterations all phenotypes existed in the population. However, at the end of deterministic reproduction A-cells have been repressed. Together with quite strong oscillations, it may indicate that the process changes frequently and may depend on cells chosen in each iteration (semi-synchronous updating). Similar results can be obtained with switching reproduction, but then the oscillations are smaller and A-cells vanish shortly after the simulation start. Quantitative reproduction gives quite a different result in comparison with other reproduction types as well as the mean-field model. Here R-cells are the dominating ones, which is contrary to the results of the mean-field model, but a bit more similar to the switching-type reproduction (which may

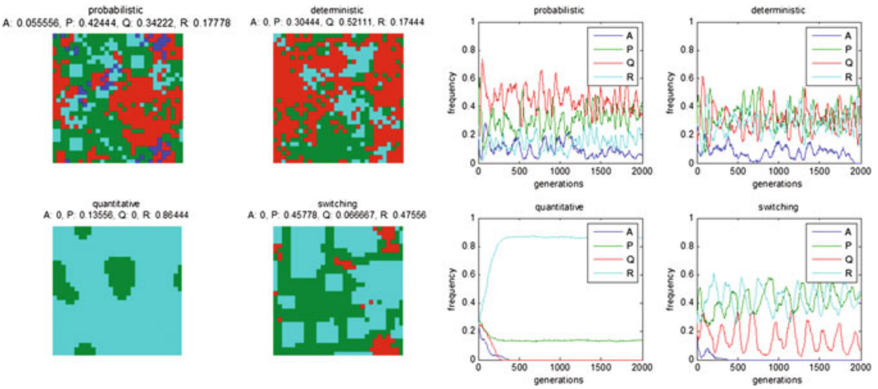


**Fig. 10.11** SEGT results for i = 0.3, j = 0.4, f = 0.4, g = 0.5, e = 0.3, h = 0.1

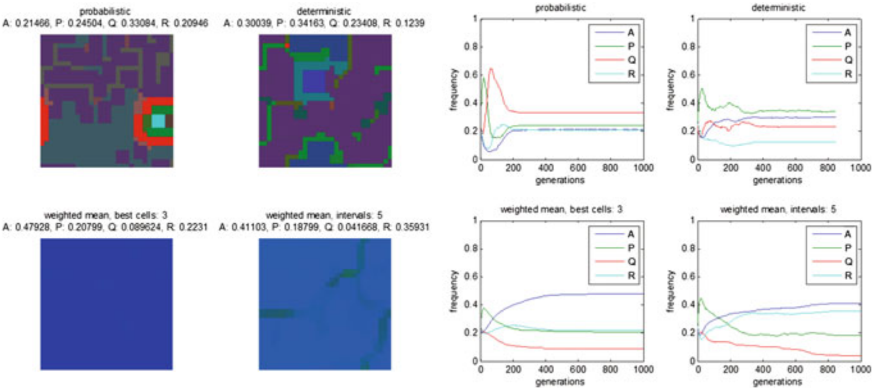


**Fig. 10.12** MSEG results for i = 0.3, j = 0.4, f = 0.4, g = 0.5, e = 0.3, h = 0.1

contain also some quantitative parts). What is more, only two phenotypes survive in the population. Figure 10.12 presents results of the MSEG approach for four reproduction schemes: probabilistic and deterministic as for classic SEGT and with factors 3 and 5 for weighted mean reproduction and intervals one, respectively.

Similarly as for SEGT, the initial lattice was generated randomly. It is not possible to compare directly the final lattices, however the generation-charts (analogous to those given by replicator dynamics in a mean-field game) are comparable. As we can see, the dynamics is more stable than in its SEGT counterpart. Probabilistic reproduction gives results quite similar to the mean-field outcome (Q-cells are the dominating ones). Other reproduction types give different results, regarding domination in the population. However, all of them show that a quadromorphic population is possible. In the case of MSEG, the final lattices show not only the different structures and clusters of phenotypes, but also different fractions of phenotypes for each, particular cell. The model yields a large but finite number of diverse results. However, the analysis is complex due to numerous parameters, intermediate relations between phenotypes and different possible scenarios, defining the game. Other strategies may be incorporated in a similar way, which increases the complexity and dependency of the analysis on massive simulations and graphical representations. Yet another possible extension resulting from mixing different Tomlinson's models consists in including an additional phenotype M (the cells producing an autocrine antiapoptotic factor). In this case it is impossible to get a stable polymorphic population with all phenotypes. For the simplex representation (phase portrait) there are points and regions that overlap each other. For instance, the point exactly in the middle may be read as a dimorphic (M and R), trimorphic (A, P and Q) population or even a population with all phenotypes (if it was possible for this model). Unfortunately, by increasing amount of phenotypes and the size of the lattice (which may be crucial for more accurate results) the simulation becomes more and more complex in terms of numerical computations.

Although the results of modeling and simulation have only quantitative meaning, they are biologically relevant. Comparing them to results of different experiments with cell lines performed by biologists cooperating with us facilitates discussion of the impact of different parameters on the development of phenomena related to interactions of the cell populations. Moreover, these results are used to plan new experiments which may explain processes still far from being recognized. It also enables study of cancer as a network society of communicating smart cells [11].

## 10.6 Conclusions and Discussion

In this chapter we have been concerned with three issues related to the process of cancer development and transition of the disease understood as propagation of molecular failures in biological network:

- The molecular mechanisms controlling and monitoring the cell cycle;
- The fragility of some signaling pathways, which may induce neoplastic transformation in cells leading to carcinogenesis;
- The role of angiogenic switch in cancer evolution and metastasis.

The main purpose of the study was to outline our own views on the issues associated with treating the cancer as a result of fault propagation in the gene—cellular network. The study is in large part a critical survey of published material including our own contribution. This review, although partly idiosyncratic, covers such major areas of cancer-related phenomena as the role of cell cycle clock, mutagenesis, avoidance of apoptosis, production of growth factors, motility and invasion, and intra- and extracellular signaling. A complete review of approaches used to study these phenomena would require a separate volume devoted solely to mathematical modeling. Therefore we have chosen to focus on only one of possible mathematical and system theoretic tools for their modeling: theory of games. More precisely, we discuss our own simulation results related to the possible dynamics and/or spatial distribution of the processes discussed in this chapter. These results are based on the theory of evolutionary games and spatial evolutionary games. Moreover, we present our original contribution to this area, resulting from a new class of population games called mixed evolutionary games.

Our study is far from being an exhaustive review, even with respect to processes related to spatial and temporal propagation of carcinogenic signals in the intracellular networks. It is strongly related to our professional experience based on our background and collaboration with biologists and clinicians. This experience direct our considerations towards system engineering mechanisms in the processes described above. Moreover, there exists a number of questions devoted to the systemic treatment of phenomena and processes related to cancer development, which are absent in our study. Cancer is a complex systemic disease, in which inherited mutations are supplemented by acquired "hits", due to chance or exposure to environmental and behavioral factors. Some of the somatic mutations are under selection (the driver mutations) but most are neutral (passenger mutations). Despite a progress in systems biology methods, it is still not clear how to distinguish them (see [13]). As we have mentioned, cancer cells prevail over regulatory circuits which subordinate them for the good of the organism. We have not attempted to answer the "very engineering" question why, instead, they switch on a long-suppressed circuitry, which allows them to function semi-autonomously, very much at the expense of the "host" organism. Some researchers suggest that it is related to the reverse evolution of those cells to the roots of multi-cellular life (see [11]). It would provide at least a partial answer to the one of the most intriguing question of propagation phenomena in biological networks: why is it so easy for cancer to break down the regulatory pathways? Yet another process, only mentioned in our study, which justifies this way of thinking, is high leveled self-organization of cancer cells, which allows them to enroll collaboration of normal cells such as fibroblasts, create their own vasculature, and organize themselves similarly as bacteria do to explore and colonize remote environments (in the metastatic process). On the other hand, the progress in the self-organization and evolutionary advantage of cancer populations is considered by other researchers as a step forward in the evolution. We hope that some of these problems could be studied with the tools presented in this chapter.

**Acknowledgments** We would like to thank Joanna Rzeszowska and Maria Wideł for their valuable biological insight and consultations. This work was partially supported by National Science Centre, Poland, grants DEC-2012/04/A/ST7/00353 (AS, JS, RJ, MK), 2011/03/B/ST6/04384 (DB). Numerical experiments and simulations were performed on the Ziemowit computational cluster (http://www.ziemowit.hpc.polsl.pl/) in the Laboratory of Bioinformatics and Computational Biology, The Biotechnology, Bioengineering and Bioinformatics Centre Silesian BIO-FARMA, created in the POIG.02.01.00-00-166/08 and expanded in the POIG.02.03.01-00-040/13 projects.

## References

1. Airley, R.: Cancer Chemotherapy. Wiley-Blackwell, New York (2009)
2. Alberts, B.: Molecular Biology of the Cell, 4th edn. Garland Science, New York (2002)
3. Anderson, A.R.A., Hassanein, M., Branch, K.M., Lu, J., Lobdell, N.A., Maier, J., Basanta, D., Weidow, B., Narasanna, A., Arteaga, C.L., Reynolds, A.B., Quaranta, V., Estrada, L.,

Weaver, A.M.: Microenvironmental independence associated with tumor progression. Cancer Res. **69**(22), 8797–8806 (2009). doi:10.1158/0008-5472.CAN-09-0437. http://dx.doi.org/10.1158/0008-5472.CAN-09-0437

4. Bach, L., Sumpter, D.J.T., Alsner, J., Loeschcke, V.: Spatial evolutionary games of interactions among generic cancer cells. J. Theor. Med. **5**, 47–58 (2003)
5. Bach, L.A., Bentzen, S.M., Alsner, J., Christiansen, F.B.: An evolutionary-game model of tumour-cell interactions: possible relevance to gene therapy. Eur. J. Cancer **37**(16), 2116–2120 (2001)
6. Baeriswyl, V., Christofori, G.: The angiogenic switch in carcinogenesis. Semin. Cancer. Biol. **19**(5), 329–337 (2009). doi:10.1016/j.semcancer.2009.05.003. http://dx.doi.org/10.1016/j.semcancer.2009.05.003
7. Ball, D.A., Adames, N.R., Reischmann, N., Barik, D., Franck, C.T., Tyson, J.J., Peccoud, J.: Measurement and modeling of transcriptional noise in the cell cycle regulatory network. Cell Cycle **12**(19), 3203–3218 (2013). doi:10.4161/cc.26257. http://dx.doi.org/10.4161/cc.26257
8. Basanta, D., Deutsch, A.: A game theoretical perspective on the somatic evolution of cancer. Selected Topics in Cancer Modeling: Genesis, Evolution, Immune Competition, and Therapy. Springer, New York (2008)
9. Basanta, D., Gatenby, R.A., Anderson, A.R.A.: Exploiting evolution to treat drug resistance: combination therapy and the double bind. Mol. Pharm. **9**(4), 914–921 (2012). doi:10.1021/mp200458e. http://dx.doi.org/10.1021/mp200458e
10. Basanta, D., Scott, J.G., Rockne, R., Swanson, K.R., Anderson, A.R.A.: The role of idh1 mutated tumour cells in secondary glioblastomas: an evolutionary game theoretical view. Phys. Biol. **8**(1), 015,016 (2011). doi:10.1088/1478-3975/8/1/015016. http://dx.doi.org/10.1088/1478-3975/8/1/015016
11. Ben-Jacob, E., Coffey, D.S., Levine, H.: Bacterial survival strategies suggest rethinking cancer cooperativity. Trends Microbiol. **20**(9), 403–410 (2012). doi:10.1016/j.tim.2012.06.001. http://dx.doi.org/10.1016/j.tim.2012.06.001
12. Bottaro, D.P., Liotta, L.A.: Cancer: out of air is not out of action. Nature **423**(6940), 593–595 (2003). doi:10.1038/423593a. http://dx.doi.org/10.1038/423593a
13. Bozic, I., Antal, T., Ohtsuki, H., Carter, H., Kim, D., Chen, S., Karchin, R., Kinzler, K.W., Vogelstein, B., Nowak, M.A.: Accumulation of driver and passenger mutations during tumor progression. PNAS **107**(43), 18545–18550 (2010). doi:10.1073/pnas.1010978107. http://dx.doi.org/10.1073/pnas.1010978107
14. Branzei, D., Foiani, M.: Regulation of DNA repair throughout the cell cycle. Nat. Rev. Mol. Cell Biol. **9**(4), 297–308 (2008). doi:10.1038/nrm2351. http://dx.doi.org/10.1038/nrm2351
15. Brosh, R., Rotter, V.: When mutants gain new powers: news from the mutant p53 field. Nat. Rev. Cancer **9**(10), 701–713 (2009). doi:10.1038/nrc2693. http://dx.doi.org/10.1038/nrc2693
16. Brosh, R., Rotter, V.: Transcriptional control of the proliferation cluster by the tumor suppressor p53. Mol. Biosyst. **6**(1), 17–29 (2010). doi:10.1039/b911416e. http://dx.doi.org/10.1039/b911416e
17. Camidge, D., Jordell, D.: Introduction to the cellular and molecular theory of cancer, chapter 24: chemotherapy. Oxford University Press, Oxford (2005)
18. Cantley, L.C., Neel, B.G.: New insights into tumor suppression: pten suppresses tumor formation by restraining the phosphoinositide 3-kinase/akt pathway. Proc. Natl. Acad. Sci. USA **96**(8), 4240–4245 (1999)
19. Cha, H.J., Yim, H.: The accumulation of DNA repair defects is the molecular origin of carcinogenesis. Tumour Biol. **34**(6), 3293–3302 (2013). doi:10.1007/s13277-013-1038-y. http://dx.doi.org/10.1007/s13277-013-1038-y
20. Chen, Y.W., Chu, H.C., Ze-Shiang, L., Shiah, W.J., Chou, C.P., Klimstra, D.S., Lewis, B.C.: p16 stimulates cdc42-dependent migration of hepatocellular carcinoma cells. PLoS One **8**(7), e69,389 (2013). doi:10.1371/journal.pone.0069389. http://dx.doi.org/10.1371/journal.pone.0069389
21. Consortium, U.: Activities at the universal protein resource (uniprot). Nucl. Acids Res. **42**(Database issue), D191–D198 (2014). doi:10.1093/nar/gkt1140. http://dx.doi.org/10.1093/nar/gkt1140

22. Cooper, G., Hausman, R.: The Cell: A Molecular Approach, 6th edn. Sinauer Associates, Sunderland (2013)
23. Crespi, B., Summers, K.: Evolutionary biology of cancer. Trends Ecol. Evol. **20**(10), 545–552 (2005). doi:10.1016/j.tree.2005.07.007. http://dx.doi.org/10.1016/j.tree.2005.07.007
24. Csikasz-Nagy, A., Kapuy, O., Toth, A., Pal, C., Jensen, L.J., Uhlmann, F., Tyson, J.J., Novak, B.: Cell cycle regulation by feed-forward loops coupling transcription and phosphorylation. Mol. Syst. Biol. **5**, 236 (2009). doi:10.1038/msb.2008.73. http://dx.doi.org/10.1038/msb.2008.73
25. de Miranda, N.F.C.C., Peng, R., Georgiou, K., Wu, C., Falk Sorqvist, E., Berglund, M., Chen, L., Gao, Z., Lagerstedt, K., Lisboa, S., Roos, F., van Wezel, T., Teixeira, M.R., Rosenquist, R., Sundstrom, C., Enblad, G., Nilsson, M., Zeng, Y., Kipling, D., Pan-Hammarstrom, Q.: DNA repair genes are selectively mutated in diffuse large b cell lymphomas. J. Exp. Med. **210**(9), 1729–1742 (2013). doi:10.1084/jem.20122842. http://dx.doi.org/10.1084/jem.20122842
26. Denayer, E., de Ravel, T., Legius, E.: Clinical and molecular aspects of Ras related disorders. J. Med. Genet. **45**(11), 695–703 (2008). doi:10.1136/jmg.2007.055772. http://dx.doi.org/10.1136/jmg.2007.055772
27. Derouiche, A., Cousin, C., Mijakovic, I.: Protein phosphorylation from the perspective of systems biology. Curr. Opin. Biotechnol. **23**(4), 585–590 (2012). doi:10.1016/j.copbio.2011.11.008. http://dx.doi.org/10.1016/j.copbio.2011.11.008
28. Dingli, D., Chalub, F.A.C.C., Santos, F.C., Van Segbroeck, S., Pacheco, J.M.: Cancer phenotype as the outcome of an evolutionary game between normal and malignant cells. Br. J. Cancer **101**(7), 1130–1136 (2009). doi:10.1038/sj.bjc.6605288. http://dx.doi.org/10.1038/sj.bjc.6605288
29. d'Onofrio, A., Gandolfi, A.: Chemotherapy of vascularised tumours: role of vessel density and the effect of vascular "pruning". J. Theor. Biol. **264**(2), 253–265 (2010). doi:10.1016/j.jtbi.2010.01.023. http://dx.doi.org/10.1016/j.jtbi.2010.01.023
30. Dvorak, H.F.: Tumors: wounds that do not heal. similarities between tumor stroma generation and wound healing. N. Engl. J. Med. **315**(26), 1650–1659 (1986). doi:10.1056/NEJM198612253152606. http://dx.doi.org/10.1056/NEJM198612253152606
31. Elias, J., Dimitrio, L., Clairambault, J., Natalini, R.: The p53 protein and its molecular network: modelling a missing link between DNA damage and cell fate. Biochim. Biophys. Acta **1844**(1 Pt B), 232–247 (2014). doi:10.1016/j.bbapap.2013.09.019. http://dx.doi.org/10.1016/j.bbapap.2013.09.019
32. Fan, Y., Mao, R., Yang, J.: Nf-kappab and stat3 signaling pathways collaboratively link inflammation to cancer. Protein Cell **4**(3), 176–185 (2013). doi:10.1007/s13238-013-2084-3. http://dx.doi.org/10.1007/s13238-013-2084-3
33. Feng, Z., Chen, J., Wei, H., Gao, P., Shi, J., Zhang, J., Zhao, F.: The risk factor of gallbladder cancer: hyperplasia of mucous epithelium caused by gallstones associates with p16/cyclind1/cdk4 pathway. Exp. Mol. Pathol. **91**(2), 569–577 (2011). doi:10.1016/j.yexmp.2011.06.004. http://dx.doi.org/10.1016/j.yexmp.2011.06.004
34. Fisher, J.C.: Multiple-mutation theory of carcinogenesis. Nature **181**(4609), 651–652 (1958)
35. Folkman, J.: Tumor angiogenesis: therapeutic implications. N. Engl. J. Med. **285**(21), 1182–1186 (1971). doi:10.1056/NEJM197111182852108. http://dx.doi.org/10.1056/NEJM197111182852108
36. Fujarewicz, K.: Planning identification experiments for cell signaling pathways—an nfkb case study. Int. J. Appl. Math. Comp. Sci. **20**(4), 773–780 (2010)
37. Fujarewicz, K., Kimmel, M., Lipniacki, T., Świerniak, A.: Adjoint systems for models of cell signaling pathways and their application to parameter fitting. IEEE/ACM Trans. Comput. Biol. Bioinform. **4**(3), 322–335 (2007). doi:10.1109/tcbb.2007.1016. http://dx.doi.org/10.1109/tcbb.2007.1016
38. Gatenby, R.A., Vincent, T.L.: An evolutionary model of carcinogenesis. Cancer Res. **63**(19), 6212–6220 (2003)

39. Gautschi, O., Ratschiller, D., Gugger, M., Betticher, D.C., Heighway, J.: Cyclin d1 in non-small cell lung cancer: a key driver of malignant transformation. Lung Cancer **55**(1), 1–14 (2007). doi:10.1016/j.lungcan.2006.09.024. http://dx.doi.org/10.1016/j.lungcan.2006.09.024
40. Gerstung, M., Nakhoul, H., Beerenwinkel, N.: Evolutionary games with affine fitness functions: applications to cancer. Dyn. Games Appl. **1**, 370–385 (2011)
41. Girard, P.M., Foray, N., Stumm, M., Waugh, A., Riballo, E., Maser, R.S., Phillips, W.P., Petrini, J., Arlett, C.F., Jeggo, P.A.: Radiosensitivity in nijmegen breakage syndrome cells is attributable to a repair defect and not cell cycle checkpoint defects. Cancer Res. **60**(17), 4881–4888 (2000)
42. Glisovic, T., Bachorik, J.L., Yong, J., Dreyfuss, G.: Rna-binding proteins and post-transcriptional gene regulation. FEBS Lett. **582**(14), 1977–1986 (2008). doi:10.1016/j.febslet.2008.03.004. http://dx.doi.org/10.1016/j.febslet.2008.03.004
43. Gluick, T., Yuan, Z., Libutti, S.K., Marx, S.J.: Mutations in cdkn2c (p18) and cdkn2d (p19) may cause sporadic parathyroid adenoma. Endocr. Relat. Cancer **20**(6), L27–L29 (2013). doi:10.1530/ERC-13-0445. http://dx.doi.org/10.1530/ERC-13-0445
44. Goel, S., Duda, D.G., Xu, L., Munn, L.L., Boucher, Y., Fukumura, D., Jain, R.K.: Normalization of the vasculature for treatment of cancer and other diseases. Physiol. Rev. **91**(3), 1071–1121 (2011). doi:10.1152/physrev.00038.2010. http://dx.doi.org/10.1152/physrev.00038.2010
45. Han, W., Yu, K.N.: Ionizing Radiation, DNA Double Strand Reak and Mutation. Advances in Genetics Research. Nova Science Publishers, New York (2010)
46. Hanahan, D., Folkman, J.: Patterns and emerging mechanisms of the angiogenic switch during tumorigenesis. Cell **86**(3), 353–364 (1996)
47. Hanahan, D., Weinberg, R.A.: The hallmarks of cancer. Cell **100**(1), 57–70 (2000)
48. Hanahan, D., Weinberg, R.A.: Hallmarks of cancer: the next generation. Cell **144**(5), 646–674 (2011). doi:10.1016/j.cell.2011.02.013. http://dx.doi.org/10.1016/j.cell.2011.02.013
49. Hanel, W., Moll, U.M.: Links between mutant p53 and genomic instability. J. Cell Biochem. **113**(2), 433–439 (2012). doi:10.1002/jcb.23400. http://dx.doi.org/10.1002/jcb.23400
50. Harris, S.L., Levine, A.J.: The p53 pathway: positive and negative feedback loops. Oncogene **24**(17), 2899–2908 (2005). doi:10.1038/sj.onc.1208615. http://dx.doi.org/10.1038/sj.onc.1208615
51. Haupt, Y., Maya, R., Kazaz, A., Oren, M.: Mdm2 promotes the rapid degradation of p53. Nature **387**(6630), 296–299 (1997). doi:10.1038/387296a0. http://dx.doi.org/10.1038/387296a0
52. Hemnani, T., Parihar, M.S.: Reactive oxygen species and oxidative DNA damage. Indian J. Physiol. Pharmacol. **42**(4), 440–452 (1998)
53. Hofbauer, J., Shuster, P., Sigmund, K.: Replicator dynamics. J. Theor. Biol. **100**, 533–538 (1979)
54. Howell, G.M., Hodak, S.P., Yip, L.: Ras mutations in thyroid cancer. Oncologist **18**(8), 926–932 (2013). doi:10.1634/theoncologist.2013-0072. http://dx.doi.org/10.1634/theoncologist.2013-0072
55. Iwamoto, K., Hamada, H., Eguchi, Y., Okamoto, M.: Mathematical modeling of cell cycle regulation in response to DNA damage: exploring mechanisms of cell-fate determination. Biosystems **103**(3), 384–391 (2011). doi:10.1016/j.biosystems.2010.11.011. http://dx.doi.org/10.1016/j.biosystems.2010.11.011
56. Iwamoto, K., Hamada, H., Okamoto, M.: Mechanism of cell cycle disruption by multiple p53 pulses. Genome Inform. **25**(1), 12–24 (2011)
57. Jain, R.K.: Normalization of tumor vasculature: an emerging concept in antiangiogenic therapy. Science **307**(5706), 58–62 (2005). doi:10.1126/science.1104819. http://dx.doi.org/10.1126/science.1104819
58. Johnson, L.N.: The regulation of protein phosphorylation. Biochem. Soc. Trans. **37**(Pt 4), 627–641 (2009). doi:10.1042/BST0370627. http://dx.doi.org/10.1042/BST0370627

59. Jonak, K., Kurpas, M., Puszyński, K.: Prediction of the Behavior of Mammalian Cells after Exposure to Ionizing Radiation Based on the New Mathematical Model of ATM-Mdm2-p53 Regulatory Pathway. Information Technologies in Biomedicine. Springer, Berlin (2014)
60. Kohn, K., Bohr, V.: Genomic Instability and DNA Repair. The Cancer Handbook. Wiley, New York (2005). doi:10.1002/0470025077.chap07
61. Król, D.: Propagation phenomenon in complex networks: theory and practice. New Gener. Comput. **32**(3–4), 187–192 (2014)
62. Krześlak, M., Świerniak, A.: Spatial evolutionary games and radiation induced bystander effect. Arch. Contr. Sci. **21**, 135–150 (2011)
63. Ledoux, A., Perkins, N.D.: Nf-kappab and the cell cycle. Biochem. Soc. Trans. **42**(1), 76–81 (2014). doi:10.1042/BST20130156. http://dx.doi.org/10.1042/BST20130156
64. Loeb, L.A.: Mutator phenotype may be required for multistage carcinogenesis. Cancer Res. **51**(12), 3075–3079 (1991)
65. Lord, C.J., Ashworth, A.: The DNA damage response and cancer therapy. Nature **481**(7381), 287–294 (2012). doi:10.1038/nature10760. http://dx.doi.org/10.1038/nature10760
66. Mansury, Y., Diggory, M., Deisboeck, T.: Evolutionary game theory in an agent-based brain tumor model: exploring the genotype-phenotype link. J. Theor. Biol. **238**, 145–156 (2006)
67. Marciniak-Czochra, A., Kimmel, M.: Reaction-diffusion approach to modeling of the spread of early tumors along linear or tubular structures. J. Theor. Biol. **244**(3), 375–387 (2007). doi:10.1016/j.jtbi.2006.08.021. http://dx.doi.org/10.1016/j.jtbi.2006.08.021
68. Maynard Smith, J.: Evolution and the Theory of Games. Cambridge University Press, Cambridge (1982)
69. Maynard Smith, J., Price, G.R.: The logic of animal conflict. Nature **246**, 15–18 (1973)
70. Merlo, L.M.F., Pepper, J.W., Reid, B.J., Maley, C.C.: Cancer as an evolutionary and ecological process. Nat. Rev. Cancer **6**(12), 924–935 (2006). doi:10.1038/nrc2013. http://dx.doi.org/10.1038/nrc2013
71. Molatore, S., Kiermaier, E., Jung, C.B., Lee, M., Pulz, E., Hofler, H., Atkinson, M.J., Pellegata, N.S.: Characterization of a naturally-occurring p27 mutation predisposing to multiple endocrine tumors. Mol. Cancer **9**, 116 (2010). doi:10.1186/1476-4598-9-116. http://dx.doi.org/10.1186/1476-4598-9-116
72. Moll, U.M., Petrenko, O.: The mdm2-p53 interaction. Mol. Cancer Res. **1**(14), 1001–1008 (2003)
73. Naugler, W.E., Karin, M.: Nf-kappab and cancer-identifying targets and mechanisms. Curr. Opin. Genet. Dev. **18**(1), 19–26 (2008). doi:10.1016/j.gde.2008.01.020. http://dx.doi.org/10.1016/j.gde.2008.01.020
74. von Neumann, J.: Theory of self producing automata. University of Illinois Press, Champaign (1966)
75. Neurath, H., Walsh, K.A.: Role of proteolytic enzymes in biological regulation (a review). PNAS **73**(11), 3825–3832 (1976)
76. Nishida, N., Yano, H., Nishida, T., Kamura, T., Kojiro, M.: Angiogenesis in cancer. Vasc. Health Risk Manag. **2**(3), 213–219 (2006)
77. Pietenpol, J.A., Stewart, Z.A.: Cell cycle checkpoint signaling: cell cycle arrest versus apoptosis. Toxicology **181–182**, 475–481 (2002)
78. Pomerening, J.R.: Positive-feedback loops in cell cycle progression. FEBS Lett. **583**(21), 3388–3396 (2009). doi:10.1016/j.febslet.2009.10.001. http://dx.doi.org/10.1016/j.febslet.2009.10.001
79. Pruitt, K.D., Brown, G.R., Hiatt, S.M., Thibaud-Nissen, F., Astashyn, A., Ermolaeva, O., Farrell, C.M., Hart, J., Landrum, M.J., McGarvey, K.M., Murphy, M.R., O'Leary, N.A., Pujar, S., Rajput, B., Rangwala, S.H., Riddick, L.D., Shkeda, A., Sun, H., Tamez, P., Tully, R.E., Wallin, C., Webb, D., Weber, J., Wu, W., DiCuccio, M., Kitts, P., Maglott, D.R., Murphy, T.D., Ostell, J.M.: Refseq: an update on mammalian reference sequences. Nucl. Acids Res. **42**(Database issue), D756–D763 (2014). doi:10.1093/nar/gkt1114. http://dx.doi.org/10.1093/nar/gkt1114

80. Puszyński, K., Hat, B., Lipniacki, T.: Oscillations and bistability in the stochastic model of p53 regulation. J. Theor. Biol. **254**(2), 452–465 (2008). doi:10.1016/j.jtbi.2008.05.039. http://dx.doi.org/10.1016/j.jtbi.2008.05.039
81. Puszyński, K., Jaksik, R., Świerniak, A.: Regulation of p53 by sirna in radiation treated cells: simulation studies. Int. J. Appl. Math. Comp. Sci. **22**(4), 1011–1018 (2012). doi:10.2478/v10006-012-0075-9. http://dx.doi.org/10.2478/v10006-012-0075-9
82. Roman, A., Munger, K.: The papillomavirus e7 proteins. Virology **445**(1–2), 138–168 (2013). doi:10.1016/j.virol.2013.04.013. http://dx.doi.org/10.1016/j.virol.2013.04.013
83. Saier, M.: Classification of Transmembrane Transport Systems in Living Organisms. Biomembrane Transport. Academic Press, San Diego (1999)
84. Shiloh, Y.: ATM and related protein kinases: safeguarding genome integrity. Nat. Rev. Cancer **3**(3), 155–168 (2003). doi:10.1038/nrc1011. http://dx.doi.org/10.1038/nrc1011
85. Sigmund, K., Nowak, M.A.: Evolutionary game theory. Curr. Biol. **9**(14), R503–R505 (1999)
86. Sinha, R.P., Hader, D.P.: Uv-induced DNA damage and repair: a review. Photochem. Photobiol. Sci. **1**(4), 225–236 (2002)
87. Śmieja, J., Jamaluddin, M., Brasier, A.R., Kimmel, M.: Model-based analysis of interferon-beta induced signaling pathway. Bioinformatics **24**(20), 2363–2369 (2008). doi:10.1093/bioinformatics/btn400. http://dx.doi.org/10.1093/bioinformatics/btn400
88. Solomon, H., Madar, S., Rotter, V.: Mutant p53 gain of function is interwoven into the hallmarks of cancer. J. Pathol. **225**(4), 475–478 (2011). doi:10.1002/path.2988. http://dx.doi.org/10.1002/path.2988
89. Steinbrunn, T., Stuhmer, T., Gattenlohner, S., Rosenwald, A., Mottok, A., Unzicker, C., Einsele, H., Chatterjee, M., Bargou, R.C.: Mutated ras and constitutively activated akt delineate distinct oncogenic pathways, which independently contribute to multiple myeloma cell survival. Blood **117**(6), 1998–2004 (2011). doi:10.1182/blood-2010-05-284422. http://dx.doi.org/10.1182/blood-2010-05-284422
90. Świerniak, A.: Combined anticancer therapy as a control problem. Advances in Control Theory and Automation, Monograph of Committee of Automatics and Robotics PAS. Commitee on Automatic Control and Robotics Polish Academy of Science (2012)
91. Świerniak, A., Kimmel, M., Śmieja, J.: Mathematical modeling as a tool for planning anticancer therapy. Eur. J. Pharmacol. **625**(1–3), 108–121 (2009). doi:10.1016/j.ejphar.2009.08.041. http://dx.doi.org/10.1016/j.ejphar.2009.08.041
92. Świerniak, A., Krześlak, M.: Game theoretic approach to mathematical modeling of radiation induced bystander effect. In: Proceedings of 16th National Conference on Applications of Mathematics in Biology and Medicine, Krynica (2010)
93. Świerniak, A., Krześlak, M.: Application of evolutionary games to modeling carcinogenesis. Math. Biosci. Eng. **10**(3), 873–911 (2013)
94. Takashima, A., Faller, D.V.: Targeting the ras oncogene. Exp. Opin. Ther. Targets **17**(5), 507–531 (2013). doi:10.1517/14728222.2013.764990. http://dx.doi.org/10.1517/14728222.2013.764990
95. Todeschini, A.L., Georges, A., Veitia, R.A.: Transcription factors: specific DNA binding and specific gene regulation. Trends Genet. 1–9 (2014) (in press). doi:10.1016/j.tig.2014.04.002. http://dx.doi.org/10.1016/j.tig.2014.04.002
96. Tomlinson, I.P.: Game-theory models of interactions between tumour cells. Eur. J. Cancer **33**(9), 1495–1500 (1997)
97. Tomlinson, I.P., Bodmer, W.F.: Failure of programmed cell death and differentiation as causes of tumors: some simple mathematical models. PNAS **92**(24), 11130–11134 (1995)
98. Tomlinson, I.P., Bodmer, W.F.: Modelling the consequences of interactions between tumour cells. Br. J. Cancer **75**(2), 157–160 (1997)
99. Tyson, J.J., Novak, B.: Functional motifs in biochemical reaction networks. Annu. Rev. Phys. Chem. **61**, 219–240 (2010). doi:10.1146/annurev.physchem.012809.103457. http://dx.doi.org/10.1146/annurev.physchem.012809.103457

100. Ulanet, D.B., Hanahan, D.: Loss of p19(arf) facilitates the angiogenic switch and tumor initiation in a multi-stage cancer model via p53-dependent and independent mechanisms. PLoS One **5**(8), e12,454 (2010). doi:10.1371/journal.pone.0012454. http://dx.doi.org/10.1371/journal.pone.0012454

101. van Moorselaar, R.J.A., Voest, E.E.: Angiogenesis in prostate cancer: its role in disease progression and possible therapeutic approaches. Mol. Cell Endocrinol. **197**(1–2), 239–250 (2002)

102. Wang, E., Lenferink, A., O'Connor-McCourt, M.: Cancer systems biology: exploring cancer-associated genes on cellular networks. Cell Mol. Life Sci. **64**(14), 1752–1762 (2007). doi:10.1007/s00018-007-7054-6. http://dx.doi.org/10.1007/s00018-007-7054-6

103. Ward, A.F., Braun, B.S., Shannon, K.M.: Targeting oncogenic ras signaling in hematologic malignancies. Blood **120**(17), 3397–3406 (2012). doi:10.1182/blood-2012-05-378596. http://dx.doi.org/10.1182/blood-2012-05-378596

104. Warfel, N.A., El-Deiry, W.S.: p21waf1 and tumourigenesis: 20 years after. Curr. Opin. Oncol. **25**(1), 52–58 (2013). doi:10.1097/CCO.0b013e32835b639e. http://dx.doi.org/10.1097/CCO.0b013e32835b639e

105. White, E.: Exploiting the bad eating habits of ras-driven cancers. Genes Dev. **27**(19), 2065–2071 (2013). doi:10.1101/gad.228122.113. http://dx.doi.org/10.1101/gad.228122.113

106. Yamazaki, D., Kurisu, S., Takenawa, T.: Regulation of cancer cell motility through actin reorganization. Cancer Sci. **96**(7), 379–386 (2005). doi:10.1111/j.1349-7006.2005.00062.x. http://dx.doi.org/10.1111/j.1349-7006.2005.00062.x

107. Zetter, B.R.: Angiogenesis and tumor metastasis. Annu. Rev. Med. **49**, 407–424 (1998). doi:10.1146/annurev.med.49.1.407. http://dx.doi.org/10.1146/annurev.med.49.1.407

108. Zhao, W., Huang, C.C., Otterson, G.A., Leon, M.E., Tang, Y., Shilo, K., Villalona, M.A.: Altered p16(ink4) and rb1 expressions are associated with poor prognosis in patients with nonsmall cell lung cancer. J. Oncol. **2012**, 957,437 (2012). doi:10.1155/2012/957437. http://dx.doi.org/10.1155/2012/957437

109. Zilfou, J.T., Lowe, S.W.: Tumor suppressive functions of p53. Cold Spring Harb. Perspect Biol. **1**(5), a001,883 (2009). doi:10.1101/cshperspect.a001883. http://dx.doi.org/10.1101/cshperspect.a001883

# Chapter 11
# Propagation Models and Analysis for Mobile Phone Data Analytics

**Derek Doran and Veena Mendiratta**

**Abstract** People in modern society use mobile phones as their primary way to retrieve information and to connect with others across the globe. The kinds of connections these devices support give rise to networks at many levels, from those among devices connected by near-field radio or bluetooth, to society-wide networks of phone calls made between individuals. This chapter introduces state-of-the-art propagation models that have been applied to understand such networks. It discusses how the models are used in many innovative studies, including how short-lived information spreads between phone callers, how malware spreads within public places, how to detect fraudulent and scamming activity on a phone network, and to predict the propensity of a user to unsubscribe from a mobile phone carrier. It concludes with a discussion of future research opportunities for the study of propagation modeling to mobile phone data analytics.

## 11.1 Introduction and Motivation

As of February 2013, an astonishing 6.8 billion mobile phone subscriptions are active across the world.[1] This huge number of subscribers, constituting a majority of the world's population, reflects how citizens of countries with varying socioeconomic conditions all rely on cellular devices to communicate and connect with others. These devices, which are typically full of data about who our contacts are, the kind of information we share, who we communicate with, and our physical location have also emerged as an attractive platform to study human behaviors and activity across large geographic regions. For example, the analysis of mobile phone data has

[1] http://mobithinking.com/mobile-marketing-tools/latest-mobile-stats.

D. Doran (✉)
Department of Computer Science and Engineering, Kno.e.sis Research Center,
Wright State University, Dayton, OH 45435, USA
e-mail: derek.doran@wright.edu

V. Mendiratta
Bell Labs, Alcatel-Lucent, Naperville, IL 60563, USA
e-mail: veena.mendiratta@alcatel-lucent.com

D. Król et al. (eds.), *Propagation Phenomena in Real World Networks*,
Intelligent Systems Reference Library 85, DOI 10.1007/978-3-319-15916-4_11

led to the development of algorithms that automatically identify physical locations people are interested in [36] and reveal the typical mobility patterns of people within a country [5, 7, 41]. Studying the structure of calls placed between mobile devices have identified strong correlations between physical location and social friendship strength [14], and have even been used to discover regional economies within developing countries [34]. Such studies highlight the amazing ways mobile phone datasets let us study the collective actions of people through the structure of people's communications, interactions, and friendships. We have only just started to tap into the intelligence that can be mined from these datasets.

The main function of a mobile phone is to transfer information from one user to another. This information may be contained in the informal and unstructured data users transmit via SMS messages and voice calls. It may also be formal, structured data like images, files, and video transferred between devices in local areas through near-field communication (NFC) and bluetooth radios, or across the Internet to our contacts through smartphone apps and other third party services. Records about these transmissions are typically stored on a mobile device and may be collected by smartphone applications running in the background, or recorded by the network service provider. These records may reveal who information was transmitted to, what type of data was transferred, where the sender physically performed the transmission, and when the data transfer occurred. The relational nature of this data naturally gives rise to *networks* of users or devices within which many kinds of information flow. Since mobile phones are now ubiquitous across the world, understanding the process through which information propagates [29] across these networks adds to our basic understanding of the modern communication patterns humans exhibit.

In this chapter, we present a number of state-of-the-art propagation models and algorithms that have been applied to networks extracted from mobile phone datasets. The methods were selected so as to demonstrate the diversity of models that have been developed for this purpose, and to highlight the way they support many different innovative applications. We first discuss models that support the study of information diffusion across society. We then present epidemiological models that are tailored to the unique dynamics of communication between mobile devices in local areas, and how they are applied to anticipate the dynamics of malware transference between devices in local-area networks. Finally, we introduce sender-specific, receiver-specific, and clustering algorithms that compute the spread of information or influence and support a host of network provider services, including the identification of scammers and to predict who is likely to switch providers in the near future. We emphasize that this chapter is not meant to be a comprehensive survey of mobile phone data analytics, nor is it meant to present an exhaustive summary of the many propagation models that have been developed and could be utilized to understand mobile phone datasets. Instead, it intends to: (i) demonstrate how modeling propagation phenomena is a critical tool for mobile phone data analytics; (ii) show researchers interested in mobile phone data analytics the kinds of propagation models and algorithms they should be equipped with; and (iii) expose a number of avenues of future research in the study of propagation within mobile phone datasets.

This chapter is organized as follows. Section 11.2 introduces the kind of data and networks that may be extracted from mobile phone communications. Section 11.3 presents propagation models used to understand the diffusion of information across mobile phone networks. Section 11.4 discusses epidemiological models and their application to the study of mobile malware. Section 11.5 introduces propagation models used in the development of novel applications for service providers. Section 11.6 reflects on the works presented and offers exciting directions for future research. Concluding remarks are given in Sect. 11.7.

## 11.2 Mobile Phone Data Analytics

We define mobile phone data analytics as the mining and analysis of datasets whose records encode communication or interaction activities between mobile phone devices. Such datasets are typically extracted from a collection of devices that individually contain information about who the device's owner (i.e. mobile phone *user*) has a relationship with, as defined by the collection of mobile phone numbers in its contact list. The devices may also carry information about when and to whom the user transmits information via NFC or bluetooth to neighboring devices, and records of the SMS messages and phone calls she placed.

Smartphone applications that have sufficient permissions to access a device's data may extract information for performing mobile phone data analytics. Because it is difficult to deploy and obtain permissions for retrieving such information, however, researchers typically rely on call data records (CDRs) provided by a mobile phone service provider. The kind of information encoded in a typical CDR is provided in Table 11.1. It includes the phone number of the caller and callee, the duration of the call, the cost of making the call, if the call was on or off the provider's network, the date and duration of the call, and the base station used to connect the caller's mobile phone to the network. The position of this base station is used in many studies to approximate the position of a user when they make a phone call, while the duration, cost, and whether the call was on network may be attributes reflecting the strength of a relationship between two individuals. For example, we may infer that the back and forth off-network calls recorded in entries 1 and 2 of Table 11.1 represent communication between users who share a strong relationship since they both incurred a financial cost and spoke for a long period of time. The `calling_num` and `called_num` fields may be used to create a directed network of mobile phone calls between users.

The data collected from a mobile device or by a service provider may capture the structure of communications and relationships at multiple levels as illustrated in Fig. 11.1. At the *local level*, mobile devices equipped with NFC or bluetooth technology are capable to transmitting data between each other. At this level, the analysis exploits the position of devices to define a structure of possible local data transmissions to discover how data propagates in a small public area. These data transmissions may correspond to the automatic pinging of neighboring bluetooth

**Table 11.1** Typical format and entries of a CDR

| Call | Base_station | Calling_num | Called_num | Start_time | Duration | Cost | On_net_call |
|---|---|---|---|---|---|---|---|
| 1 | nyc-1234 | 8881112234 | 9992223345 | 01/01/2014 14:35:23 | 38 | 3.80 | FALSE |
| 2 | paris-2512 | 9992223345 | 8881112234 | 01/03/2014 18:35:23 | 100 | 10.00 | FALSE |
| 3 | chicago-3412 | 8882345678 | 8883345722 | 01/03/2014 18:40:30 | 50 | 0.00 | TRUE |

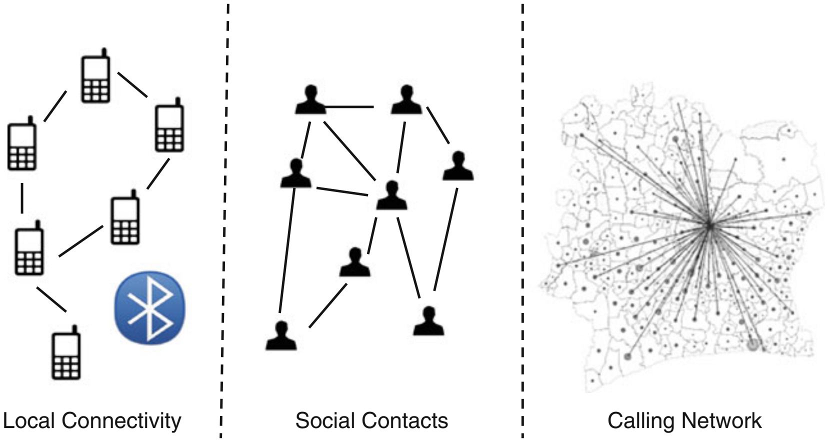


**Fig. 11.1** Structure within mobile phone datasets among devices (local level), address books (contact level), and network-wide communication (calling level)

devices for deriving the density of people in an environment or to infer real-life social networks [37], data transmissions by an intentionally installed application, or the automatic spreading of malware or viruses that run without the user knowing [51]. At the *contact level*, phone numbers collected from the address book of users' devices are extracted and aggregated to form a collection of social relationships among users. At the *calling level*, CDRs collected by service providers may be used to study human communication across large geographic areas.

Many different propagation models and algorithms are applied to mobile phone data at the local, contact, and calling level. We divide the models covered in this chapter according to the type of analytics they support in Fig. 11.2, namely by: (i) understanding information diffusion; (ii) modeling malware propagation; and (iii) supporting network provider applications. These three types represent the diversity of the different kinds of mobile phone analytics supported by propagation models. They range from academic studies that seek to discover intrinsic qualities about information dissemination, to theoretical analyses that can be used to solve a widely-applicable problem facing society, to models that are specifically developed to support a business enterprise.

A roadmap of the specific models presented in this chapter is listed in Table 11.2, including a brief summary of the model and the network level it operates on. Information diffusion studies rely on structural models that capture spreading dynamics (causality trees), statistical approaches for characterizing complex distributions (mixture models and correlation metrics), and algorithms for finding users who play a critical role in the diffusion process (user clustering). The analysis of malware uses carefully designed SIR, SIS, and SIDR epidemiological models that also incorporate the unique mobility dynamics of mobile phone devices in public spaces. Practical applications utilize user clustering algorithms and new models for energy propagation across a mobile phone network.

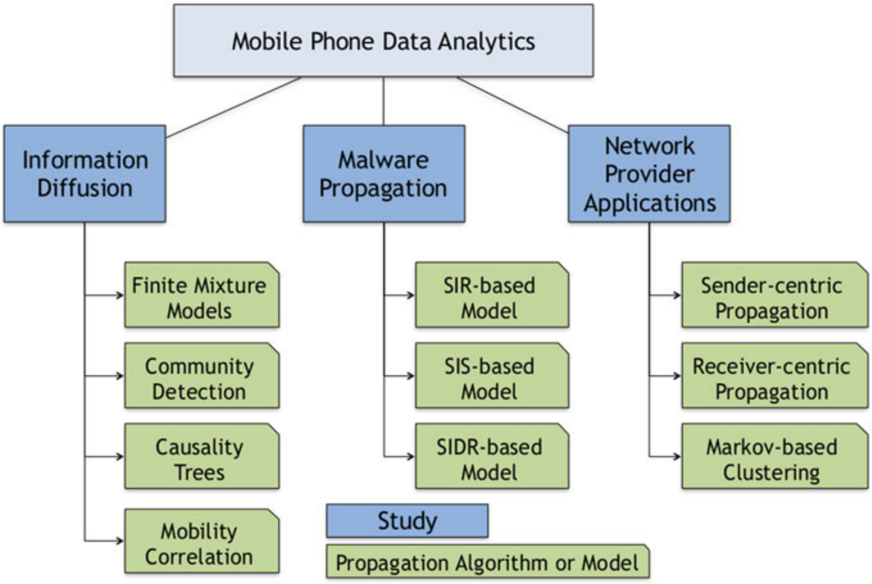


**Fig. 11.2** Roadmap of the propagation models and the studies they support in this chapter

## 11.3 Information Diffusion

No matter the medium used to transmit information between mobile phone devices and their users, the chance that information spreads from one user to another depends on the strength of the relationship they share and on the dynamic nature of the information as it passes through the network of mobile users. Intuitively, the strength of the relationship shared between two users strongly impacts when, how often, and what kind of information is shared. Calls to a family member, for example, may happen much more frequently compared to calls made to a bank or doctor’s office, increasing the chance of meaningful information dissemination. The dynamic nature of different types of information as it passes through a network is also critical. For example, information about topical news stories may experience a large number of transmissions due to the ‘buzz’ surrounding breaking news, but the frequency of these transmissions may decay over time as this news becomes less relevant. As another example, a person may broadly share a major life event with all of their contacts, but share a more personal story to a small subset of her contacts. We next examine information propagation models and algorithms that incorporates either of these aspects to make discoveries about the nature of mobile phone communication patterns.

**Table 11.2** Propagation models and methods in this chapter

| Propagation model | Summary | Structure level |
|---|---|---|
| *Information diffusion* (Sect. 11.3) | | |
| Finite mixture model | Model the distribution of times between and duration of diffusions between links | Network |
| Mobility correlation metrics | Search for relationships between physical mobility and creating new network connections | Network |
| Causality trees | Models *pass-along dynamics* where information transmissions can only happen within a window of time $\tau$ | Contact |
| Community based greedy algorithm | Identify most influential members of a phone network under a weighted influence diffusion model | Network |
| *Malware propagation* (Sect. 11.4) | | |
| SIR-based model | Models malware spreading dynamics where devices can recover and immunize themselves from infection | Local |
| SIS-based model | Model steady-state infection levels of malware in local areas where devices cannot be immune from infection | Local |
| SIDR-based model | Optimize the maximum damage that may be caused by a malware epidemic that not only infects but also kills devices | Local |
| *Novel applications* (Sect. 11.5) | | |
| Sender-centric energy propagation | Model accumulation of influence where senders force information on receivers | Network |
| Receiver-centric energy propagation | Model accumulation of influence where receivers deicide what information is retained | Network |
| Markov clustering algorithm | Discover fraudulent users based on the structure of information propagation | Network |

### *11.3.1 Characterizing Diffusion Frequency: Finite Mixture Models*

One of the most basic properties of communication patterns are the frequency with which transmissions are made between users. Kim et al. [26] performed a comprehensive analysis of these frequencies by analyzing the communication activity of over one million bi-directional pairs of mobile phone subscribers from a nationwide cellular provider. Using metadata about each subscriber, they classified pairs by whether they are both in-network, if they are in different networks (out-network), and if they are family members. The objective of their study is to develop a universal model that can accurately capture the frequency of information exchange across all three classes of users, as characterized by the inter-arrival times between calls made between pairs.

An initial analysis by the authors revealed that the empirical distribution of inter-arrival times do not follow a single exponential distribution, suggesting that the call arrival process is not Poisson for at least one class of pairs. They thus propose a finite mixture model to universally characterize the inter-arrival times of all pairs. A mixture model assumes that the data is drawn from a finite number of $K$ distributions as specified by:

$$f(\mathbf{y}; \psi) = \prod_{i=1}^{n} f(y_i, \psi) = \prod_{i=1}^{n} \sum_{k=1}^{K} w_k f_k(y_i; \theta_k) \tag{11.1}$$

where $f_k(y_i; \theta_k)$ is one of the $K$ distributions of the mixture, $\mathbf{y} = (y_1, \ldots, y_n)$ is the vector of observations, and $w_1, \ldots, w_k$ are positive mixing weights assigned such that $\sum_{k=1}^{K} w_k = 1$. They decide to consider mixture models of Gamma, Lognormal, and Gaussian distributions because they all are capable of modeling non-negative random variables with a large range of possible density shapes. The model's collection of parameters $\psi$ can be estimated by the expectation-maximization algorithm and use the Akaike Information Criterion [27] and Minimum Description Length [4] metrics to find the best number of components $K$.

#### 11.3.1.1 Model Application

The authors fitted Gamma, Lognormal, and Gaussian finite mixture models to the distribution of inter-arrival times across all pairs of users and within the three different types of pairs. Although each pair of users exhibit a unique calling pattern, they find that the lognormal mixture model offers a very tight fit (MSE = $0.3605 \times 10^{-4}$). Family pairs were found to require a mixture model that is of higher order for fitting their inter-arrival time distributions, but of lower order to fit their call duration distribution. In contrast, out-of-network pairs need a low order mixture model to capture inter-arrival times and high order model to capture call durations. About 27 % of all pairs' inter-arrival time distributions are best fitted by a single order model.

### *11.3.2 User Mobility and Diffusion: Mobility Correlation Metrics*

The distribution of peoples' physical locations are intimately related to the way information diffuses among users of a mobile phone network. This is because, practically, information passed through mediums like NFC or bluetooth require devices to be near each other. Furthermore, sociological studies confirm how we are more likely to connect and share information with those near us because the social links encouraging this behavior are driven by spatial proximity [43]. Understanding the way humans diffuse physically is thus an important consideration when studying the spread of information across a mobile phone network.

Call data records record the id of the cell phone tower used by a sender and receiver used during a conversation. By mapping these id's to the physical position of the tower, we can study the approximate locations where a user regularly submits mobile phone calls and their daily trajectories through a geographic area. We can also find correlations between the physical proximity of two users and frequency of calls made between them. Such correlations can be expressed using a variety of metrics proposed by Wang et al. [48]:

1. *Distance*. This metric refers to the most likely physical distance separating two users in the network. Let $L_i(x)$ be the location of user $x$ during his $i$th recorded call and $n(x)$ be the total number of calls made by $x$. Let

$$PV(x, l) = \sum_{i=1}^{n(x)} \mathbb{1}(l = L_i(x))/n(x) \tag{11.2}$$

   be the probability that a user $x$ visits a location $l$ where $\mathbb{1}(q)$ is an indicator function that returns 1 if the statement $q$ evaluates to true and 0 otherwise. The *most likely* location of user $x$ is thus given by $ML(x) = \arg\max_{l \in Loc} PV(x, l)$. We can define the distance $d$ between users $x$ and $y$ as $d(x, y) = \text{dist}(ML(x), ML(y))$ where dist is a measure of geographic distance.
2. *Spatial Co-location rate*. This metric captures the likelihood that two users visit in the same location but not necessarily at the same time. Assuming their visits are independent, it is given as:

$$CoL(x, y) = \sum_{l \in Loc} PV(x, l) \times PV(y, l) \tag{11.3}$$

   where $Loc$ is the set of locations that both $x$ and $y$ have been recorded as visiting.
3. *Cosine similarity*. This metric uses cosine similarity to capture how similarly two users frequent the same locations. It is given as:

$$Cos(x, y) = \sum_{l \in Loc} \frac{CoL(x, y)}{||PV(x, l)|| \times ||PV(y, l)||} \tag{11.4}$$

4. *Weighted cosine similarity*. This metric corresponds to the *tf-idf* version of cosine similarity. In essence, the *tf-idf* version adds weight to co-location events within low-density areas, that is, areas where users are seldom seen, and penalizes high-density areas. For example, pairs that frequent seldom visited locations may be more likely to have a relation than those who both frequent common locations.
5. *Co-location rate*. This metric measures the probability two users will be located in the same location in the same day and hour. It is given as:

$$CoL = \frac{\sum_{i=1}^{n(x)} \sum_{j=1}^{n(y)} \theta(\Delta T - |T_i(x) - T_j(y)|)\mathbb{1}(L_i(x) = L_j(y))}{\sum_{i=1}^{n(x)} \sum_{j=1}^{n(y)} \theta(\Delta T - |T_i(x) - T_j(y)|)} \tag{11.5}$$

where $\theta(x)$ is the Heaviside step function and $\Delta T = 1$ h. The numerator counts the number of times two users visit the same location at the same time, normalized by how frequently they are active at the same time.

6. *Weighted Co-location rate*. This is the *tf-idf* version of *CoL* where the normalization factor is the log of the number of users at each location in the same hour.
7. *Extra-role Co-location rate*. This metric is defined by *CoL* taken over only evening and weekend hours. Co-location during these times may be an important predictor of an offline relationship.

#### 11.3.2.1 Model Application

Wang et al. applied these mobility correlation metrics to a dataset consisting of over 6 million users and 90 million calls [48]. Their analysis focuses on the 50,000 most active individuals in the dataset. They find that the geographical distances between pairs exhibit a heavy-tailed distribution, which is consistent with a number of previous findings [28, 31, 33]. The CoL and SCoL measures of co-location rates reveal how many pairs can be found to be visiting the same locations, but for short periods of time. Furthermore, the geographical distance between two users decays only logarithmically with the *Col* and *Cos* measures of proximity.

Since mobility and information diffusion are intimately related to each other, the authors utilize these metrics to predict whether new information diffusions will occur in the future. They train a C4.5 decision tree to classify whether a potential connection in the calling network that does not exist during time period $t$ will emerge at time period $t + 1$. The tree is trained with network structure and mobility correlation metrics and yields a precision of 73.5 % and recall of 66.1 %. Compared to classifiers that only consider network structure metrics, this precision and recall is an order of magnitude higher. This confirms that human mobility patterns are intimately associated with the future diffusion of information across new connections.

### *11.3.3 Modeling Pass-Along Dynamics: Causality Trees*

An intriguing type of information people share between both their peers and close contacts are breaking news stories or rumors. We define such information to be *short-lived*, as people become disinterested in news and rumors the longer it has been since it broke out on the network. To model the dissemination of such information, we consider *pass-along spreading processes* [39]. A pass-along spreading process is defined as one where a user can only pass information to some subset of their contacts, and only within a short a period of time $\tau$ since she received the information. This

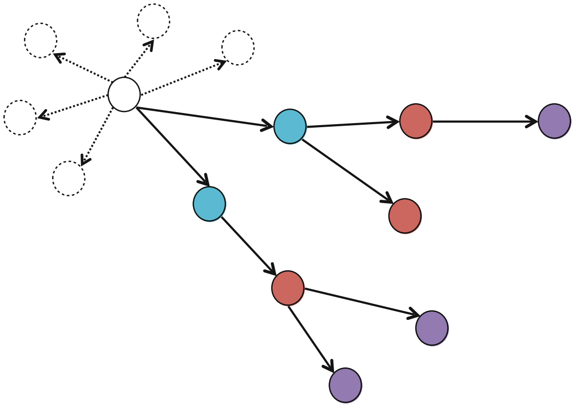


**Fig. 11.3** Example of a pass-along dynamic modeled by a causality tree where $d = 3$ and $s = 9$. The root user of the tree passes information along to $k_o = 2$ out of his $k'_o = 7$ contacts. Users with the same color exist at the same depth of the tree

pass-along process repeats for every user that has received this information, until no new users have become informed. Figure 11.3 illustrates how a pass-along process is modeled as a diffusion tree whose depth $d$ corresponds to the maximum distance from the initiator to an informed user, size $s$ is the number of users who become informed, and whose paths represent a sequence of consecutive communications whose time between calls are always less than or equal to $\tau$.

A causality tree can be used to model the probability a user $k$ will be contacted by $k_i$ other users and subsequently pass along information to $k_o$ users within a given $\tau$. Such an event corresponds to a user in a causality tree that has in-degree $k_i$ and out-degree $k_o$ given $\tau$. These probabilities can be used to identify the extent to which a user in the network chooses to participate in the pass-along process. For example, a user who is entirely disinterested in spreading information would be represented in the model as a user in the tree with large in-degree and low out-degree. Users excited to pass information widely corresponds to those having large out-degrees in the cascade tree. Let $k'_i$ and $k'_o$ be the in- and out-degree of node $k$ across a network of contacts (e.g., the number of others who have $k$ as a contact and number of contacts $k$ has, respectively). Since $k$ receives and sends information from and to only a subset of all contacts during a pass-along along process, the probability $k$ has in-degree $k_i$ and out-degree $k_o$ in a causality tree is given by:

$$\begin{aligned} p(k_i, k_o; \tau) = & \sum_{k'_i = k_i; k'_o = k_o}^{\infty} p_\infty(k'_i, k'_o) \\ & \times \binom{k'_i}{k_i} T_i(k'_i, \tau)^k (1 - T_i(k'_i, \tau))^{k'_i - k_i} \\ & \times \binom{k'_o}{k_o} T_o(k'_o, \tau)^k (1 - T_o(k'_o, \tau))^{k'_o - k_o} \end{aligned} \tag{11.6}$$

where $p_\infty(i, o)$ is the probability of finding a node with in-degree $i$ and out-degree $o$ in the contact network and $T_o(k_i', \tau)$ ($T_i(k_i', \tau)$) is the probability that a user will send (receive) information to (from) $k_i'$ ($k_o'$) users within $\tau$ time. We can simplify Eq. 11.6 by assuming that the number of users $k$ chooses to send information to is independent of the number of sources $k$ received the information from, so that $T_i(k_i', \tau) = T_o(k_o', \tau) = T(k, \tau)$. If we assume that the frequency with which calls are made over a communication link follow a Poisson process [47], we can model the probability that $k$ will send short-lived information to a contact within $\tau$ time as $1 - \exp(-\rho\,\tau)$, where $\rho$ is defined as the sending rate of $k$. Thus, we can define $T(k, \tau)$ as:

$$T(k, \tau) = \int d\rho\, p(\rho)(1 - \exp(-\rho\tau)) \tag{11.7}$$

where $p(\rho)$ is the probability density of user sending rates across the network.

While $p(k_i, k_o; \tau)$ represents the dynamics of individuals in a pass-along process, statistics about the causality tree itself sheds light into the overall reach and participation of users sharing short-lived information. The recursive nature of a cascade tree can be exploited for this purpose. For example, to compute the probability of observing a tree with size $s$ $p(s; \tau)$, we begin by defining the probability of finding a tree of size $s = 1$ by $p(s = 1; \tau) = p(k_o = 0; \tau)$, i.e., the probability of a tree whose root node has out-degree zero. $p(s = 2; \tau)$ can then be defined as the probability that a root node has out-degree 1 and its child node also has out-degree 1. We can continue to extend this definition recursively to define all $p(s'; \tau)$ for $s' < s$. This recursive relationship may be expressed by the generating function $G(z, \tau) = E(z^s) = \sum_{s=1} p(s; \tau)z^s$, which obeys the self-consistency equation:

$$G(z; \tau) = zg(1, G(z; \tau); \tau) \tag{11.8}$$

where $g(1, y; \tau)$ is the generating function for the probability a user in a cascade tree has out-degree $k_i$:

$$g(1, y; \tau) = \sum_{k_o} p(k_o; \tau)y^{k_o} \tag{11.9}$$

The cascade size distribution can thus be found by taking derivatives of the generating function:

$$p(s = n; \tau) = \frac{1}{n!}\frac{\partial^n G(z; \tau)}{\partial z^n}|_{z=0} \tag{11.10}$$

A similar recursive formulation can be used to model the probability a tree has depth $d$ $p(d; \tau)$. Let $E_d(\tau)$ be the probability a causality has some depth less than or equal to $d$. This probability obeys the relation:

$$E_d(\tau) = g_1(E_{d-1}(\tau); \tau) = g_d(0; \tau) \tag{11.11}$$

where $g_1(y; \tau) = g(1, y\,\tau)$ and $g_n(y; \tau) = g_1(g_{n-1}(y; \tau); \tau)$. Then the probability of a tree having depth $d$ is given as:

$$p(d; \tau) = E_d(\tau) - E_{d-1}(\tau) = g_d(0; \tau) - g_{d-1}(0; \tau) \tag{11.12}$$

#### 11.3.3.1 Model Application

Peruani et al. proposed the propagation model based on causality tree presented above [39]. They applied it to a mobile phone dataset from a European telecom with 1,044,397 users that made 13,983,433 calls between them. They derive the parameters of the model from the dataset, and identify a very close fit between the modeled cascade size and probability distributions with the observations they make in the original dataset.

The model's application draws a number of findings about the nature of pass-along dynamics in a mobile phone network. Specifically, they find the existence of super-spreaders and receivers, who are giant hubs that absorb or widely disseminate information along the network. They also discover that pass-along dynamics are extremely sensitive to the correlation of users' in- and out-degree distributions. Furthermore, at large time-scales ($\tau$), the spreading dynamics actually become dominated by correlations in the topological structure of users in the network, not the pass-along process. In other words, pass-along processes only capture the dynamics of information exchange at a very local level (e.g. to degree 1 or 2-neighbors).

### *11.3.4 Diffusion Maximization: Community Based Greedy Algorithm*

A third-party wishing to influence as many people as possible may wish to find $k$ seed nodes who can maximize the spread of their influential information across the network. These seed nodes represent *influential users*, defined as those who share information with the intention of changing another's personal opinions or beliefs. If an influencer is successful, newly influenced people subsequently pass their information off through their set of connections, and so forth. Influence propagation thus exhibits the same pass-along dynamics modeled by causality trees, but without a time constraint. In other words, an influencer may try to sway another at any time, regardless of the time passed since they themselves became influenced. The extent to which influence propagates through a network thus depends only on the position and number of influencers that begin the diffusion process.

Although finding the $k$ users to initially influence such that the maximum number of others on the network become influenced is an NP-hard problem, greedy algorithms are capable of finding an approximate solution to within a factor of $(1 - 1/e - \varepsilon)$ [8, 21, 32], the algorithms are too inefficient to process very large mobile phone networks. Instead, community-based greedy algorithms that identify the top-$k$ most influential nodes in a mobile phone network have been proposed as a way to efficiently solve this problem [49]. We first define the *diffusion speed* of information from user $v_i$ to $v_j$ in the network as:

$$\lambda_{ij} = 2\bar{\lambda}\frac{w_{ij}}{w_{max} + w_{min}} \tag{11.13}$$

where $\bar{\lambda}$ is the empirically measured average calling rate of users in a network and $w_{ij}$ is the weight of the directed connection from $i$ to $j$. These weights should correspond to a quality of the connection such that the higher its value, the faster the rate of information diffusion. For example, the number of calls or SMS messages sent between the users could correspond to a connection weight. The algorithm then considers the following diffusion process:

1. Select a set of active seed nodes $S_0$ active at an initial time $t = 0$.
2. Increment the time clock to $t = t+1$. Choose a node $v_i$ from the set $S_{t-1}$. For every directed neighbor $v_j$ of $v_i$, try to influence her with probability $\lambda_{ij}$. If successful, add $v_j$ to the set $S_t$.
3. Update $S_t = S_t \cup (S_{t-1} \setminus v_i)$.
4. Repeat steps 2 and 3 until the set of active nodes $S_t = \emptyset$.
5. The set of all nodes influenced by the seed set $S_0$ is given as $\mathcal{V}_S = \bigcup_{i=0}^{t-1} S_i$. Define the *degree of influence* of $S_0$ to be $R(S_0) = \mathcal{V}_S/N$ where $N$ is the number of users in the mobile phone network.

Under this process, we can efficiently find a set of seed nodes $S$ such that $|S| = k$ if we assume that the mobile phone network can be divided into many communities of users. A community is a set of users who frequently communicate with each other and are more likely to be swayed by information originating within it. If information originating from one community will have almost no influence over the members of another, a good approximation for finding the top influencers of the network is to find users *within individual communities* that maximize the spread of influence within them. The algorithm for finding these communities are given in [49].

Let $I_k$ be the set of the $k$ seed users that leaves the strongest amount of influence on the network. To find $I_k$, assume that we already have constructed the set $I_{k-1}$ thus far. We define by how much the degree of influence across the network will increase by adding the most influential member within community $C_m$ to the set $I_k$ as:

$$\Delta R_m = \max\{R_m(I_{k-1} \cup v_j) - R_m(I_{k-1})|v_j \in C_m\} \tag{11.14}$$

Thus, we can choose the $k$th influential user to add to $I_{k-1}$ by choosing the most influential member in the community that has the largest $\Delta R_m$ value. $\Delta R_m$ can be found using any previously proposed less efficient algorithm to find the most influential user within a small network [8]. This less efficient algorithm is expected to perform in a reasonable amount of time since it only runs across a community of the entire calling network.

We can use dynamic programming to efficiently choose the community from which an influential user is added $I_k$. Let $R[m, k]$ be the influence degree yielded if the $k$th most influential user is selected from one of the first $m$ communities. Then,

$$R[m, k] = \max(R[m-1, k], R[m, k-1] + \Delta R_m) \tag{11.15}$$

where $R[m, 0] = 0$ and $R[0, k] = 0$. In other words, if a user from the first $m-1$ communities yields a smaller influence degree than choosing the most influential user from community $m$, choose it from $C_m$. Otherwise, choose it from one of the $m-1$ former communities. The choice of these former communities is represented by $s[m, k]$. It is given by:

$$s[m, k] = \begin{cases} s[m-1, k], & R[m-1, k] \geq R[M, k-1] + \Delta R_m \\ m, & R[m-1, k] < R[m, k-1] + \Delta R_m \end{cases} \tag{11.16}$$

with $s[0, k] = 0$.

#### 11.3.4.1 Model Application

Wang et al. presented and applied this community-based greedy algorithm to a network of SMS messages between 723,201 users collected by a major telecom company [49]. Under many choices of $K$ and $\bar{\lambda}$, the community-based method was able to find a set of users that yields the largest spread of influence in the network compared to many previously proposed algorithms. It has modest run-times (on the order of thousands of seconds) under the entire range of parameter settings used for experimental analysis under a simple hardware configuration (2.0 GHz Xeon 8 Core CPU; 8GB Memory; Debian 4.0 Operating System). Experimental analysis finds that the improvement in influence degree rises exponentially fast with $\bar{\lambda}$ (the average rate of diffusion). Influence degree increases just logarithmically with $K$, with very small gains for $K > 15$. The study also finds that approximately $M = 25$ communities offers the best tradeoff between minimizing computation time and maximizing influence degree. In summary, the method demonstrates how a small number of influencers ($\sim$15) are sufficient to widely disseminate influence across society-wide communication networks. Furthermore, numerous latent communities exist within a mobile phone network, where members are likely to influence each other.

## 11.4 Malware Propagation

Security and network researchers envision mobile phones as being the next frontier for malware [9, 10, 15] due to the many vulnerabilities present in mobile platforms [20], the un-savvy users operating mobile devices, and the private and valuable information they store on them. A 2011 Mobile Threats Report by Juniper Networks Mobile Threat Center found a 155 % increase in mobile malware over the past year [45]; by the end of the same year McAfee Labs had collected over 75 million samples of mobile malware. Malware is capable of changing mobile phone configurations, spamming SMS messages, dialing pay-to-call numbers, and collecting private information stored on the device.

Understanding the development of malware on a mobile phone network, and devising techniques to combat this threat, require novel propagation models. This is because these always-on devices may be susceptible to infection through local NFC or bluetooth transmissions , by connecting to a compromised public access point, or through a compromised link shared across a contact network via SMS [38]. These infections may thus quickly propagate through a mobile network as it infects and transmits from device to device. In many ways, this is analogous to the spread of an infectious disease through a population of people who congregate in public places. Thus, many researchers have proposed different variations of common epidemiological models (e.g. SI [3], SIR [23], SIS [24]) to better understand the spreading dynamics of mobile phone malware, and to propose methods that thwart their spread. This section details some of these recent models and methods, and discusses their application to mobile phone networks.

### *11.4.1 Infection Dynamics with Recoverable Devices: SIR Epidemiological Model*

Rhodes et al. introduced an extended SIR epidemiological model for modeling the spread of malware opportunistically shared between bluetooth enabled smartphones [42]. The model considers not just the rate at which devices become **susceptible** (**S**), **infected** (**I**), or **recovered** (**R**), but also the rate at which devices come into contact with each other and the devices' transmission profiles. We first assume that mobile devices are spatially distributed over a fixed region with density $\rho$. Each individual device moves independently of all others with constant velocity $v$. If any device moves within the transmission radius $R$ of another device in the area, the devices make contact and there is an opportunity for malware to spread. Thus, a new individual device that moves with its own velocity $v_i$ will be exposed to contact by device $i$ during a time period $dt$ if it lies within a rectangular-shaped area that is covered by the movement of $i$ and lies in the direction of the vector $w = v_i - v$. The total area covered by $i$ during $dt$ is given by $dA = 2Rwdt$ where $w = \sqrt{v_i^2 + v^2 - 2v_i \cos\phi}$ is the relative speed of the device and $\phi$ is the angle between velocity vectors. Thus, the number of devices in transmission range of $i$ is given by:

$$\gamma = \int_0^{2\pi} \frac{dN_\phi}{dt} = \frac{\rho R}{\pi} \int_0^{2\pi} wd\,\phi \tag{11.17}$$

This reduces to:

$$\gamma = \frac{4\rho R}{\pi}(v_i + v) \int_0^{\pi/2} \left(1 - \frac{4vv_i}{(v+v_i)^2}\sin^2\omega\right)^{1/2} d\omega \tag{11.18}$$

If we make the simplifying assumption that the new device $i$ moves with the same velocity as all other devices (so that $v_i = v$), we can write Eq. 11.18 as an elliptic integral and use its standard form to find:

$$\gamma = \frac{8}{\pi}\rho vR \tag{11.19}$$

If a single device transmits malware to another within its range with probability $p$, the infection rate of devices in the system is $\beta = p\gamma$.

The model also considers a radial decay function to compute the probability a susceptible device becomes infected. The choice of a radial decay function is based on the fact that the longer a device spends in the transmission range of an infected user, the higher its chance of becoming infected, and the closer one device is to another, the longer it will take for them to be out of transmission range. Thus, we compute the probability a device at position $r$ gets transmitted malware by computing the path length between $r$ and contact with an infected node given by $2(R^2 - r^2)^{1/2}$, multiply it by the probability of infection upon falling in transmission range $p$, and normalize by the total transmission range:

$$p(r) = \frac{p}{R}(R^2 - r^2)^{1/2} \tag{11.20}$$

Integrating over all positions $r$ and substituting $p$ and $R$ for $p(r)$ in the formulation of $\beta$, we get:

$$\beta = \frac{8}{\pi}\rho v \int_0^R p(r)dr \tag{11.21}$$

which solves to:

$$\beta = 2R\rho vp \tag{11.22}$$

Using this new infection rate, we apply the SIR model to specify a malware outbreak by the differential equations:

$$\frac{d\mathbf{S}}{dt} = -\beta \frac{\mathbf{SI}}{N} \tag{11.23}$$

$$\frac{d\mathbf{I}}{dt} = \beta \frac{\mathbf{SI}}{N} - \delta \mathbf{I} \tag{11.24}$$

$$\frac{d\mathbf{R}}{dt} = \delta \mathbf{I} \tag{11.25}$$

where $I$ is the number of devices infected, $S$ is the number of susceptible devices, and $N$ is the total number of devices on the network.

#### 11.4.1.1 Model Application

The authors compared the output of the SIR-based model to a simulation of an outbreak of malware in a setting with a device density of 3000 devices/km$^2$, mean velocity of 2 km/day, transmission probability $p = 0.1$, transmission range of 5–40 m per device, and with a recovery rate of 1 device per 5 days. They find that the epidemic dynamics are mostly caused by the aggregation of many dyadic interactions, rather than spreading the malware to multiple devices at once due to the the small transmission range of the devices. However, as transmission radius increases, the SIS-model comes to a much stronger agreement with the simulation results. They conclude that the dynamics of malware propagation are greatly affected by the characteristics of the devices and of the environment they operate under. When malware that devices can recover from are transmitted over far-reaching channels, the SIS-model captures its infection dynamics very well.

### *11.4.2 Infection Dynamics Without Immunization: SIS Epidemiological Model*

Mickens et al. developed an extension of the Kephart-White (KW) epidemiological model [22] that also considers the mobility of devices within a constrained area [35]. This is an SIS (Susceptible–Infected–Susceptible) epidemiological model where devices may cycle between **susceptible** and **infected**. In other words, a device can never be completely immune and may become infected again once cured.

The traditional KW model assumes a homogeneous network topology in which all devices have a similar number of neighbors $\bar{k}$. If $I$ is the fraction of devices infected at a particular moment in time, the KW model describes the propagation of an infection as the differential equation:

$$\frac{dI}{dt} = \beta \bar{k} I(1 - I) - \delta I \tag{11.26}$$

where $t$ is the current time, $\beta$ is the propagation rate of malware from one device to another, and $\delta$ is the rate at which any infected device is cured. Holding these rates and $\bar{k}$ constant, this equation has a steady state solution of:

$$I = 1 - \frac{\delta}{\beta \bar{k}} \tag{11.27}$$

Thus, we require

$$\beta \bar{k} > \delta \tag{11.28}$$

for an infection to persist in the network. These parameters can be mapped to model the spread of mobile device malware by letting $\bar{k}$ be the average number of devices within communication range of any other device, $\beta$ be the probability a malware infected device transmits it to a health neighbor during a time period $\Delta t$, and $\delta$ be the probability an infected device cures itself during time $\Delta t$. However, extensive analysis by the authors confirm that the KW model does not accurately model the dynamics of malware that spreads by NFC or bluetooth transmissions in a local mobile phone network. This is because the homogeneity assumption held by the KW model is broken by the fact that mobile devices move around a region and have a limited transmission radius. The number of neighbors a device has at any given time is thus constantly in flux and should not be represented by a constant value $\bar{k}$. Furthermore, the KW model does not incorporate parameters for the velocity of mobile devices within an area, which they find to be a major factor in how quickly malware spreads in their simulations.

To extend the KW model, the authors consider the spatiotemporal dynamics of devices within a large rectangular area using a random waypoint mobility model. In this mobility model, devices randomly select a destination point, travel there, pause for a constant time $t_p$, and then choose another random destination point. The waypoints are independently chosen prior to departing. The speed at which devices move between waypoints is given as a random velocity chosen uniformly within some pre-specified range. Under this mobility model, the spatial density function of devices over a square region is given as:

$$S(x, y) = \frac{p_p}{a^2} + (1 - p_p)\frac{36}{a^6}(x^2 - \frac{a^2}{4})(y^2 - \frac{a^2}{4}) \tag{11.29}$$

where $a$ is the length of a side of the square region and $p_p = t_p/E[T]$ where $E[T]$ is the average time a node takes to move from one waypoint to another. Thus, if a device is at position $(x_i, y_i)$, we can derive the probability that it is within communication range of another device by the integral:

$$c(x_i, y_i) = \int_{y_i - r}^{y_i + r} \int_{x_i - \sqrt{r^2 - (y - y_i)^2}}^{x_i + \sqrt{r^2 - (y - y_i)^2}} S(x, y)dxdy \tag{11.30}$$

where $r$ is the radius of communication for all devices. We use $c$ to find the probability a device at $(x_i, y_i)$ has $k_i$ devices within communication range by:

$$Pr(x, y, k = k_i) = \binom{N-1}{k} c(x, y)^k (1 - c(x, y))^{N-k-1} \tag{11.31}$$

The expected probability two devices will be within communication range of each other is thus:

$$\bar{c} = \frac{\int_{-a/2}^{a/2} \int_{-a/2}^{a/2} c(x, y)dxdy}{a^2} \tag{11.32}$$

and the probability any device will have $k_i$ devices in communication range across the entire region is:

$$Pr(k = k_i) = \frac{\int_{-a/2}^{a/2} \int_{-a/2}^{a/2} Pr(x, y, k = k_i)dxdy}{a^2} \tag{11.33}$$

To consider mobility under the KW model, the connectivity fluctuations induced by mobility need to be incorporated. We can do so by considering the average travel time of a device from one waypoint to another $E[T]$ as a queue or pipe that takes $E[T]$ time to traverse. If the probability a device at any location has $k_i$ neighbors is $Pr(k = k_i)$, the amount of time it spends with $k_i$ neighbors while moving from one location to another is given by $E[T] \times Pr(k = k_i)$. For example, $E[T] \times Pr(k = k_0)$ is the amount of time a device has no neighbors while it travels from one destination to another, and hence can be subjected to malware cures. Otherwise, for $E[T] \times Pr$ $(k = (k_i > 0))$ time units, the device is subject to an infection pressure proportional to $\beta\, k_i$ and a cure pressure proportional to $\delta$. The extended KW model thus requires

$$\sum_{k_i=0}^{N-1} \beta\, k_i Pr(k = k_i) E[T] > c\, \delta\, E[T] \tag{11.34}$$

for a malware outbreak in the network to exist, where $c$ is a constant account for global factors affecting connectivity. Since Eq. 11.33 tells us the percentage of time a device has $k_i$ other neighbors, the total number of devices with $k_i$ neighbors across the local area is given by $N \times Pr(k = k_i)$ where $N$ is the number of devices in the local network.

To help compute the steady-state infection level of the mobile network, let us assume that the stretches of time a node has $k_i$ neighbors are large relative to the unit of time used to measure infection rates $\Delta t$. Consider a collection of $N$ queues $\{Q_{k_i}\}$, each of which initially has $N \times Pr(k = k_i)$ devices in it. When a device enters $Q_{k_i}$, it spends $E[T] \times Pr(k = k_i)$ time in it before exiting. Each queue can be thought of as a separate KW process described by the rates of infection $\beta$ and curing $\delta$, where all devices in the queue have the same $\bar{k} = k_i$ neighbors. Treating all devices in the

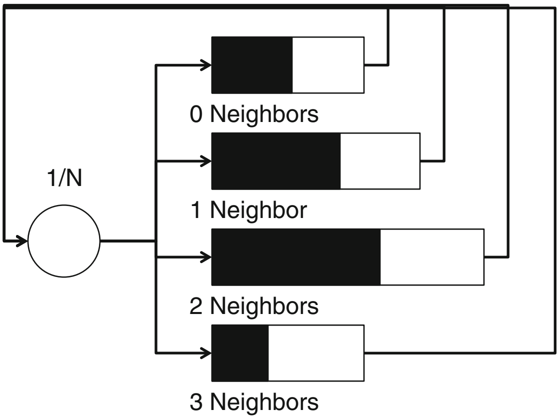


**Fig. 11.4** Queueing network for finding steady-state infection levels. Each queue is loaded with devices that have the same number of neighbors, so queue $i$ starts with $N \times Pr(k = k_i)$ devices. A random proportion of devices in queues (shown in *black*) are infected. At every time-step, we infect and cure devices according to a KW process that runs separately within each queue. After $E[T] \times Pr(k = k_i)$ time-steps, a device in queue $i$ departs and is divided into $1/N$ units. These small units are then distributed across all of the queues

same queue under the same KW process is intuitive because they all have the same number of $k_i$ neighbors, which is a core assumption of the KW model.

We can utilize a network of these queues, illustrated in Fig. 11.4, to find the steady-state infection levels. We initially place $N \times Pr(k = k_i)$ in each queue and assign a random proportion $I_{init} \in [0, 1]$ of its devices to be infected with malware. The model then iteratively updates itself in increments of $\Delta t$. At each update, it first simulates a propagation of the malware in each queue $Q_{k_i}$ using the KW equation:

$$\frac{dI_{Q_{k_i}}}{dt} = \beta\, k_i I_{Q_{K_i}}(1 - I_{Q_{K_i}}) - \delta\, I_{Q_{K_i}} \tag{11.35}$$

Every $\Delta t$ time units, the model checks if the exit time of any device has exceeded the current time, and if so, it removes the device from its queue, divides it into $N$ equally sized pieces, and enqueue's one of these pieces into the rest of the queues. Finally, every queue updates its infected percentage $I_{Q_{k_i}}$ to reflect its newly enlarged population and infection percentages. At any moment during this process, the total number of infected devices in the network is given by:

$$\sum_{k_i=0}^{N-1} I_{Q_{k_i}} \times |Q_{k_i}| \tag{11.36}$$

where $|Q_{k_i}|$ is the number of devices in a queue. The steady state number of infected devices can be found by continuing to iterate the model until these values converge.

#### 11.4.2.1 Model Application

Mickens et al. simulated a mobile device network where devices have a 100 m communication radius and move within a square region with 1000 m sides. Using various device velocities and number of devices in the network, they compare the predicted proportion of infections given under the KW model and their extended queueing based model against the simulation results. They find that the steady-state infections projected by the KW model was different from the simulation by 12.5 %, while the queueing model was only off by 4.0 %. Their analysis also discovers that epidemics are *unstable* under many parameter settings. For example, in five simulation runs lasting 200,000 s, one epidemic died out almost immediately, another lasted the entire time, and the others lasted between one- and three-fourths of the total simulation time. Thus, while the extended KW model accurately predicts average levels of infection, it hides the instability of the malware propagation process.

Finally, the authors apply their model to a scenario where the spatial distribution of devices across the region is strongly skewed, that is, where devices tend to favor specific areas within the region. This scenario may better reflect real-life mobility patterns, as users tend to congregate around popular landmarks within a region. The waypoint mobility model was modified so that nodes have a higher probability to travel to one of three 'hot-spots' in the square region. For different values of $N$, the queueing model outperforms the KW model in predicting the steady-state infection levels under the modified mobility model, but the relative improvement is not as large. The authors hypothesize that their queueing based model, which only captures the number of neighbors a device has at any time, does not necessarily capture the spatial distribution of devices within a geographic region.

### *11.4.3 Maximizing Malware Damage: SIDR Epidemiological Model*

Khouzani et al. propose the analysis of an SIDR epidemiological model to estimate the maximum amount of damage malware can impart on a local mobile wireless network [25]. They define damage as a cumulative function that increases with the number of devices that may be *infected* or *dead*. Their model allows this damage function to be generally defined, and assumes that the malware wishes to maximize damage subject to specific constraints on the energy consumption of its host devices.

Under an SIDR model, devices may fall under one of four states: **susceptible** (**S**), where an unprotected device is not yet infected; **infective** (**I**), where a device has been loaded with malware, and may propagate it to others, but the malware has not yet attacked the device; **dead** (**D**), where the malware successfully compromised the device; and **recovered** (**R**), where an updated device is immune from the infection. We let $n_\alpha(t)$ be the number of devices in state $\alpha \in \{S, I, D, R\}$ such that $\sum_\alpha n_\alpha(t) = N$ is the number of devices in the model, and the proportion of all devices in each state

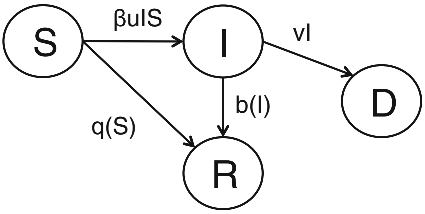


**Fig. 11.5** Markov model and transition probabilities for device states under the SIDR model

as $S(t), I(t), D(t)$, and $R(t)$ respectively so that $S(t) + I(t) + D(t) + R(t) = 1$. We assume that an outbreak begins at time $t = 0$ with the infection of $I(0) = I_0$ devices. The initial conditions of the system are $R(0) = D(0) = 0$ and $S(0) = 1 - I(0)$.

Infections occur as devices within a region $A$ move with velocity $v$. Infective devices transmit malware once they fall within a given transmission range. The probability of an infection is based on two factors: the *density* of devices within $A$, given as $\nu_1 = |N|/|A|$, and the rate at which a given pair of devices contact each other, given as $\nu_2 = 1/A$ [18]. If $u(t)$ is the product of an infected device's transmission range and rate at which it scans for devices to transmit to, the process of malware transmissions from an infected to susceptible device can be modeled by an exponential random process whose rate at time $t$ is $\hat{\beta}u(t)$ where $\hat{\beta} = \nu_1\nu_2$. Infected devices will be killed after an exponentially distributed random amount of time with rate $v(t)$. An infected or susceptible device may also recover after infection by healing or immunizing itself with rates given by $B(I(t))$ and $Q(S(t))$, respectively. The rate functions $B$ and $Q$ can be defined in any way the modeler would like, as long as they meet the following criteria: (i) $\lim_{x\to 0} B(x) < \infty$ and $\lim_{x\to 0} Q(x) < \infty$; (ii) for $0 < x < 1$, $B$ and $Q$ are positive and differentiable; and (iii) $xB(x)$ is a concave non-decreasing function of $x$ and $xq(x)$ is also a non-decreasing function of $x$.

Under these infection and recovery dynamics, we can model the rates at which devices transition between states using the continuous time Markov chain in Fig. 11.5. We represent the state vector of this chain as $V = (n_S(t), n_I(t), n_D(t))$, dropping $n_R(t)$ since $n_S(t) + n_I(t) + n_D(t) = 1 - n_R(t)$. Let $\beta = \lim_{N\to\infty} N\hat{\beta}$, $q(S) = Q(S)S$, and $b(I) = B(I)I$. According to [30], $S(t), I(t)$, and $D(t)$ will converge to the solution of the following differential equations as $N$ grows:

$$\frac{dS(t)}{dt} = -\beta\, u(t)I(t)S(t) - q(S(t)) \qquad S(0) = 1 - I_0 \tag{11.37}$$

$$\frac{dI(t)}{dt} = \beta\, uI(t)S(t) - b(I(t)) - v(t)I(t) \qquad I(0) = I_0 \tag{11.38}$$

$$\frac{dD(t)}{dt} = v(t)I(t) \qquad D(0) = 0 \tag{11.39}$$

These equations satisfy $0 \leq S(t), I(t), D(t)$ and $S(t) + I(t) + D(t) \leq 1$ for all $t$.

We now consider an attacker who wants to infect a local area in such a way that the amount of damage caused by the malware infection during a window of time

$[0, T]$ is maximized. Since damage corresponds to both the infection and killing of devices in the network, the damage function can take the following general form:

$$J = \kappa D(T) + \int_0^T f(I(t))dt \tag{11.40}$$

$\kappa$ is a positive 'reward' per device killed and$f$ is an increasing convex function where $f(0) = 0$. An attacker will try to maximize $J$ by regulating two parameters of the malware: the rate at which it will kill devices $v(t)$ and the product of the malware's transmission range and scanning rates $u(t)$. The choice of parameters for these values are subject to:

$$0 \leq v(t) \leq v_{max} \tag{11.41}$$

$$0 \leq u_{min} \leq u(t) \leq u_{max} \tag{11.42}$$

$$\int_0^T h(u(t))dt \leq C \tag{11.43}$$

The upper bound on $v(t)$ represents an inherit maximum speed at which a device can be killed by an infection. The bounds on $u(t)$ represent maximum transmission rates caused by the physical properties of an environment. The integral constraint over $h(u(t))$ ensures that the malware infection does not fully deplete an infected device's power, which it relies on to spread the infection and to eventually kill the device. It is assumed that $h$ is a non-decreasing and non-negative function. Once the malware chooses $v$ and $u$, the Markov chain's state vector $V$ will be specified at all times $t$, allowing us to solve the system of differential equations and hence compute the damage $J$ of the attack. We can then find optimal functions that control the killing, transmission, and scanning rates $v(t)$ and $u(t)$ to maximize $J$.

#### 11.4.3.1 Model Application

Khouzani et al. studied the proposed SIDR model and damage function under various parameter settings to gain insights about the malware infection and recovery processes [25]. From the optimal forms of $v(t)$ and $u(t)$, they discover how malware should start with a small killing rate that gradually increases over time. This way, infected devices are given an opportunity to infect others before being killed off. When the time window is almost over, however, devices should adopt a high killing rate to take as many down as possible. Furthermore, the malware should not decrease an infected devices transmission and scanning rates until approximately one third of the time window has elapsed. If the network can increase the recovery $B(t)$ and immunization rates ($Q(t)$ and $B(t)$) of devices, the malware must extend the period during which its transmission and scanning rates are highest ($u(t) = u_{max}$).

The authors also find a relationship between recovery rates and the total damage imposed by malware. Interestingly, they find that the amount by which damage is reduced decreases exponentially with the rate of recovery. However, with larger recover rates comes larger bandwidth and power costs for the devices.

## 11.5 Novel Applications

Mobile service providers collect a wealth of information about their customers and their calling behaviors. Hidden within these records are patterns that may be exploited to help the provider offer better service to their customers, or to make discoveries that may eventually lead to financial gains. For example, a simple analysis may reveal calling towers that are used very frequently, yet are associated with dropped calls and degraded service. Such towers should be given a higher priority for maintenance, before customers within its range decide to change providers as a result of poor service. As another example, users who receive an extraordinarily large number of calls may be targeted for a deeper investigation, to see if the number is being used as a calling center or for some other inappropriate purpose.

Beyond looking for outliers or correlations in a dataset, advanced data analytics are also utilized to find more sophisticated patterns to answer more challenging questions. In this section, we present novel propagation models used in such advanced analytics that predicts the likelihood that a customer will soon *churn*, or move to a different service provider and identifies fraudulent activity in a calling network. Churning is a significant problem for service providers because, in today's society where nearly everyone has a mobile phone, it has become very expensive to attract customers who do not yet have a phone to join their service. Furthermore, today's users are more informed about the kind of devices, the quality of the service, and the perks offered by the providers. Such providers must thus devote a significant amount of effort towards customer *retention*, rather than *acquisition*. Fraudulent activity in a calling network relates to voice-related security threats where users may reveal sensitive or private information through social engineering techniques and by calling international phone numbers. These calls carry a financial cost to both the subscribers and the service provider.

### *11.5.1 Churn Prediction: Sender-Centric Energy Propagation*

The decision to drop a service provider is based not only on a user's own satisfaction with the service, but may also be the result of social pressures from friends, family, and other close contacts who have already decided to churn. Researchers have thus turned to *energy propagation* models across the calling network of a mobile phone provider, where energy refers to information that may persuade another user to churn. In this model, users marked to have churned during a month is seeded with an amount of

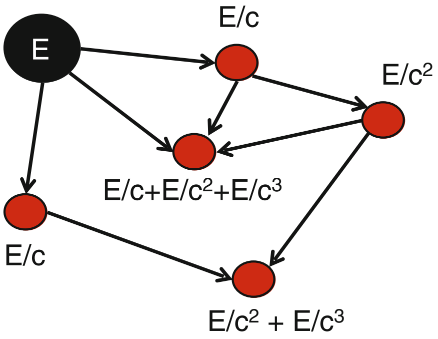


**Fig. 11.6** Generic illustration of sender-centric energy propagation

energy $E$. These churners divide this energy into smaller portions and disseminates it across all of their connections. Users who receives portions of this energy then replicates it, divides it into even smaller portions, and spreads the energy across its contacts. This process of accumulating, dividing, and spreading energy repeats until the fraction of energy received at any user drops below some threshold $t$. Figure 11.6 illustrates this spreading process. The churner (black node) distributes $E/c$ energy to its three contacts, where $c$ is some positive constant. These three contacts store this energy and then replicate a fraction $E/c^2$ of it to be sent to each of its own contacts. The total energy accumulated by a user may thus represent the likelihood that she will soon churn from the service provider.

Rather than having every user propagate a constant fraction $1/c$ of its energy to others, we define a *transfer function* $F(c)$ that returns what proportion of stored energy is transferred to each of a user's contacts. This transfer function is defined by the *sender* of the influence, putting them in control of how much energy each recipient will be exposed to. Because the receivers have no choice but to accept the energy it receives and pass it along, we refer to this energy propagation model as being **sender-centric**. Sender-centric propagation models may differ in the way senders choose what contacts to receive, and by how $F(c)$ is defined.

Dasgupta et al. proposed the following sender-centric energy propagation model for churn prediction [11]. Consider a diffusion process where at each time step $t$ there is a set of active users $X$ whose members $x \in X$ have energy $E(x, t)$. At time step $t+1$, every active user in $X$ transfers a fraction of its energy to all of their neighbors $y$. The fraction of energy sent is a function of two parameters: the spreading factor $d$ and transfer function $F$. $d$ is a constant that lets the modeler decide by how far the energy propagation should spread. Low values $d$ keep the process very local, while high values of $d$ lets energy spread far away from the churner. $F$ should be designed in a way that reflects the relative 'strength' a connection to one contact is over another, so that more energy is transferred over stronger connections. For example, information shared by a good friend who one has strong connections to will be given higher consideration. If $W_{xy}$ is the strength of a connection from $x$ to $y$, $F$ may be defined as:

$$F = \frac{W(x, y)}{\sum_{\{(x,s)|s \in N(x)\}} W(x, s)} \tag{11.44}$$

The set of active users at time $t + 1$ is then given by the set of nodes who received energy. The energy propagation process terminates at time $t^*$ if no new nodes are exposed at time $t^*$ or if the amount of energy any node is exposed to falls below a threshold value $E_T$.

#### 11.5.1.1 Model Application

Dasgupta et al. use the above sender-centric model to predict churners in a mobile call graph [11]. They define connection strength as $W_{xy} = 2/(1 + e^{-c_{xy}}) - 1$ where $c_{xy}$ is the total number of calls placed from user $x$ to $y$. They then select a threshold energy value $T_c$, where any user on the network that collects more than $T_c$ energy is predict to become a churner. They investigate the fraction of all churners correctly caught as $T_c$ decreases to include a larger fraction of users on the network. They find that the set of users having the 10 % largest amounts of energy contain approximately 45 % of all churners in a given month. From the perspective of a mobile phone service provider this is a strong result. For example, the provider can invest in a marketing campaign that targets just 10 % of its subscribers with discounts, in an attempt to prevent almost half of all potential churners from switching service providers. By comparison, the 10 % most probable churners labeled by a decision-tree classifier that uses features about the frequency a user utilizes her mobile phone service and her connectivity contains only approximately 40 % of all churners.

### *11.5.2 Churn Prediction: Receiver-Centric Energy Propagation*

In a sender-centric energy propagation model, the transfer function $F(c)$ is defined as a function of some features about the sender of information. However, one may hold the philosophic belief that it is the receiver of information, rather than the sender, who ultimately decides the degree to which she becomes influenced. This idea gives rise to an alternative class of energy propagation models that are *receiver-centric*. The rules that govern a receiver-centric propagation process may be summarized as follows [40]:

1. A user who receives energy by a neighbor will decide what proportion should be retained. This retention should be proportional to the strength of the relationship between the receiver and the sender.
2. A user only retains energy originating from a churner once.
3. However, users retain energy many times if the energies originate from different sources.

4. When a user receives some amount of influence by a neighbor, she chooses the proportion to be retained. She subsequently replicates and transmits this proportion to every one of her neighbors.

The first rule ensures that the total influence retained by a receiver will be grounded in the relationship held between the receiver and sender. The second rule captures the idea that, if a user is exposed to energy from the same source but at different iterations of the propagation, she will only retain energy from the first exposure. Intuitively, multiple exposures of energy originating from the same source would contain same information, which the receiver already considered during her first exposure. The information or influence contained in energy sent from distinct sources, however, is unique. Hence, in the third rule, a receiver is allowed to retain energy multiple times if the source of the energy is distinct. Finally, the receiver will transmit a copy of all energy she retains to all of her contacts. Her contacts will then independently decide how much energy they should retain.

Phadke et al. introduce a receiver-centric model for predicting churners in a mobile phone network [40]. They define a strength for the relationship between users $X$ and $Y$ using a vector of calling attributes $(x_1, \ldots, x_n)$. Each attribute $x_i$ is normalized by dividing it by $|x_i|$, where $|x_i| = \sqrt{\sum_{k=1}^{d} x_{ik}^2}$ so that they are of unit length. For a relationship $k$, let $k = \alpha_1\, x_1 + \alpha_2\, x_2 + \cdots + \alpha_n\, x_n$ be the weighted sum of its normalized attributes. The strength of the relationship between $X$ and $Y$ can be defined by any monotonically increasing function of $k$; the authors use $W_{XY}(k) = 1 - e^{-k/\varepsilon^2}$. This exponential function is based on the idea that when a strong relationship is established between two users, there is a higher likelihood that the information or influence within the energy passed along that connection will be retained by the receiver. $\varepsilon$ is a tunable parameter that controls the degree to which the strength of a relationship is affected by the magnitude of it's attributes.

The model computes the total amount of energy received by a user in an iterative process. It begins with the passing of $E$ energy from every node that churned in the previous month to all of its neighbors. Let $N_i$ be the set of neighbors of node $i$. A neighbor $j$ of a churner $i$ will choose to retain

$$E_j = \frac{W_{ij}}{\mathbf{W_j}} E_i \tag{11.45}$$

where $\mathbf{W_j}$ is the sum of the strength of all relationships $j$ is a part of and $E_i$ is the energy contained by churner $i$. These neighbors will then pass $E_j$ units to its neighbors, and so forth, until the number of iterations exceeds a threshold value (in their study, they terminate the process after three iterations). After the process terminates, each receiver adds together all of the energy it received.

#### 11.5.2.1 Model Application

Phadke et al. apply the receiver-centric model to a dataset of calls placed between over half a million users and churners during a two month period. For a single month of data, the authors compute the strength of each connection using the call time, number of calls made, and neighborhood overlap as relationship features. They tuned the weights $\alpha_i$ empirically in order to maximize the predictive accuracy of the propagation model. They then consider a boosted decision tree ensemble classifier that uses the amount of energy retrained along with features such as whether a contract has ended, the number of days a user is connected, the number of calls made to churners, and the charged rate for making phone calls to assign each users a probability that they will churn in the subsequent month. They find that without the energy feature, the classifier finds 35 % of all future churners among the top 10 % most likely users predicted to churn. By adding the energy accumulated, this percentage rises to approximately 40 %. In summary, they find the receiver-centric energy propagation model to be a viable alternative to a sender-centric model.

### *11.5.3 Isolating Fraudulent Activity: Markov Clustering Algorithm*

Jiang et al. [19] present a method for identifying fraudulent activity performed over voice calls in a cellular network by analyzing the structure of a calling network. Their method is rooted in the following features about fraudulent activity on mobile phone networks: (i) callers on a phone network seeking to commit fraud tend to contact a large number of people and will attract more victims to call fraudulent numbers compared to a typical user of the phone network; and (ii) fraudsters may utilize many international phone numbers at once to distribute their scheme, which lets them increase the number of victims that can be reached. This activity may be represented by observing the same set of domestic users (victims) who all call the same set of foreign (fraudulent) numbers.

These two features suggest that fraudulent activity may be characterized by finding *community structures* containing large numbers of international calls to the same collection of phone numbers. To find these communities, the method uses the Markov Clustering Algorithm (MCL). This algorithm finds communities by iterating over two steps: network *expansion* and *inflation*. At iteration $i$, the expansion step takes the square of the adjacency matrix of the network to simulate the probability of random walks of length $i + 1$ that start and terminate at every user in the call graph. In the inflation step, the elements of the squared adjacency matrix are raised to a power $\beta$, and then the matrix is scaled diagonally so that the resulting adjacency matrix is Markovian. In essence, the inflation step modifies the probabilities associated with random walks in a way that favors more probable walks. As the process repeats, matrix entries corresponding to links in low probability walks will converge to zero, so the converged adjacency matrix will only contain connections in high probability walks. The connected components of this converged matrix correspond to community structures.

To find communities that contain fraudsters, the method looks for 2-by-2 bi-partite cliques from domestic to international numbers. These 2-by-2 bi-partite cliques are the smallest structural unit that corresponds to fraudulent activity, where a set of victims who do not know each other both call the same two fraudulent numbers. The method filters out all communities that do not exhibit at least $\alpha$ bi-partite cliques of any size that have at least $\gamma$ victims.

#### 11.5.3.1 Model Application

Jiang et al. use the MCL-based method to analyze a dataset of all international voice calls made within the voice network of a major service provider [19]. They take two sources of user reports to build a ground truth list of fraudulent calls, referred to as an international revenue share fraud (IRSF) list: (i) numbers reported by customers to the provider's customer care center; and (ii) a list of phone numbers tied to customer complaints that were posted online in blogs, social media, and forums [1, 50]. They run the MCL detection algorithm on different months of data (Jan–May 2011) to study the expected lag that will occur between when fraudulent activity occurs and when it will be reported in the IRSF or online list of fraud numbers. They choose $\alpha = 5$ and $\gamma = 10$ after observing that these settings filter out over 98 % of the subgraphs while capturing over 90 % of all communities that exhibit fraud. They compare the numbers in these fraud communities against a list of over 24,000 numbers fraudulent numbers covered in the IRSF lists. They find that the extracted communities only contain 11 % of the numbers in the list. However, these 11 % of numbers attract phone calls from 85 % of all victims, and are the root cause of 78 % of all fraudulent calls in the network. Furthermore, when the authors exclude dormant numbers in the IRSF list (numbers not yet utilized or advertised by fraudsters), the detection rate increases from 11 to over 50 %.

The authors also evaluate whether the MCL algorithm can be used to identify fraudsters early, before they are reported or recorded on an IRSF list. For all fraud numbers contained in the communities extracted, the gap between the month it was extracted from in the data and the month it was added to the IRSF list is compared. For more than 80 % of the fraud numbers, the detection method precedes the user reports and in more than 60 % of these cases, the fraud numbers are discovered at least one month sooner than when a report is shared by a user.

#### 11.5.3.2 Summary of Findings

The models presented in this chapter found a number of important characteristics and new findings about mobile phone communication networks. We summarize these findings next.

- **Diffusion processes are governed by heavy-tailed distributions**. The distributions of how long information propagates between two users, and the frequency of these propagations, are characterized by mixtures of Lognormal distributions.
- **Physical co-location is strongly correlated with the formation of future connections** Users that propagate information between each other are likely to be co-located for brief periods of time. Whether or not two users exist in the same location strongly predicts whether they will form new connections in the future.
- **Short-lived information over calling networks does not diffuse widely**. The total number of others that receive short-lived information is strongly correlated with the in- and out-degree distribution of the users participating in the diffusion process. Propagations of short-lived information are generally limited to a very local level and do not spread far and wide across a calling network.
- **Epidemiological models are a flexible tool to understand local-level interactions and the spreading of malware**. Epidemiological models have been used to successfully model the dynamics of malware that spreads at local levels. Different kinds of models can incorporate specific properties of mobile devices, including the range of their transmissions and energy constraints. SIP-based models become less accurate if transmissions can only be performed devices are within very close proximity. SIS-based models may be used in scenarios where devices can never become immunized. SIDR-based models work under scenarios where devices can be killed or disabled by malware. To maximize damage, malware should wait for infections to spread before killing devices. As the recovery rates of devices increase, the total damage of a malware outbreak drops exponentially.
- **Energy propagation models can help identify future churners**. Irrespective of whether a modeler uses a sender-centric or receiver-centric propagation model, we can identify a large proportion of future churners by the total energy or influence they accumulate from past churners. Both sender- and receiver- centric propagation models offer promising results.
- **Finding user communities with bi-partite cliques can identify fraudulent activity**. Bi-partite cliques may correspond to users who send calls to the same subset of fraudulent phone numbers on the network. 80 % of the communities found through a Markov clustering algorithm containing such bi-partite cliques include fraudulent numbers not yet been reported by users.

## 11.6 Future Research Directions

The state-of-the-art propagation models presented in this chapter represent significant advances in mobile phone data analytics. However, many opportunities remain where researchers may build off of, extend, and use the discoveries made by these methods to propose new kinds of models.We next present a small sampling of these research opportunities.

1. **Marry structure and decisions in the diffusion of information**. The propagation models reviewed in this chapter concentrate on either the *structure* of a

diffusion process or on how individuals *decide* what information should be saved. For example, causality tree models only reason about the probability that certain subsets of a user's connections will be transmitted information within a given time period. Epidemiological models also rely on the structure of the network as users' devices form connections by their spatiotemporal dynamics within a local area. Sender- and receiver-centric energy propagation models, however, simply assume that information spreads widely across all connections. They then concentrate on modeling the process of deciding to retain information, including who makes the decision (sender or receiver) and how that decision is made.

More faithful models of information diffusion should simultaneously consider both structure and decision-making. For example, one should not assume that churners will decide to submit all of their contacts to peer influence. Furthermore, a receiver of short-term information spreading through a causality tree may decide to not propagate the news further if she is disinterested in the information, if her social relationship with the sender is weak, or if she does not believe that her set of contacts would be interested in the information.

2. **Explore the tradeoffs between sender- and receiver-centric propagation**. For the churn prediction problem, both sender- and receiver-centric models have been demonstrated to be similarly successful. Yet these two model types are underpinned by two very different philosophies: one asserts that the person who sends information controls how much the receiver absorbs, while the other believes that the receiver of information individually decides how much they will accept. One kind of model may be more applicable than the other depending on the setting. For example, marketing studies have demonstrated the persuasive effect that a strong advertisement [46] or speaker [44] can have on the amount of information retrained by others. On the other hand, peoples' experiences and knowledge also modulate the amount of information they choose to retain [16]. The settings under which either a sender- or receiver-centric propagation model is more appropriate remains an open question. Hybrid models that integrate both sender and receiver effects may be an effective development.
3. **Build new epidemiological models that operate on other network levels**. Epidemiological models have mostly been applied to local level networks. Although the analogy between the exchange of information among devices that are physically close and the exchange of diseases between people makes applications at the local level intuitive, the spread of information and data need not be restricted by the proximity of devices. For example, there now exist compromised applications that may submit spam messages and fraudulent links to other contacts in a person's address book [2]. Epidemiological models that operate at the contact level may suitably represent the spread of such SMS spam. Furthermore, the spread of rumors and lies across a calling network may be thought of as a systemic spread of mis-information that convinces or (infects) gullible (susceptible) individuals on the network. Thus, an epidemiological model operating at the calling level may characterize the spread of mis-information by accounting for a user's propensity for believing and spreading false information.

4. **Recognize the differences between devices**. Mobile phone devices are built with hardware that supports a variety of technological features. For example, as of 2014, only Android handsets with NFC chips built in are capable of spreading malware to other devices over this medium. Furthermore, devices that either have SMS messaging disabled or cannot support receiving them will not be able to receive information that spreads across this medium. It is thus necessary to consider the heterogeneous mix of devices with varying capabilities within propagation models over mobile phones. Furthermore, differences between devices are not only associated with hardware configurations, but also by their brand. For example, recent intriguing results have found Apple iPhone users to have more connections to others on average, and are more likely to be connected with an iPhone than an Android user [6]. Thus, at the contact level, there may be a higher propensity for information to propagate from one device to another.
5. **Integrate social features**. Ultimately, contact and calling level networks formed out of mobile phone data are *social networks* where the ties users have with many others correspond to offline relationships. Numerous methods in the literature exist to extract the social qualities of such relationships. For example, analysis of ego-network structures can identify users exhibiting egocentric or selfish tendencies [12] as well as those who sport different kinds of social roles [17]. Depending on these roles and tendencies, a user may exhibit different behaviors in a propagation model. For example, egocentric individuals who will speak with everyone simply to be noticed may send new information to all of their contacts, irrespective of whether that information is fact or fiction. Or perhaps users that lie on the periphery of two communities may decide to not let information move from one to another, out of consideration that the other community may be disinterested. We should also consider social features as we assign weights corresponding to the strength, and hence amount of information that propagates, across connections. For example, we know that exceptionally strong and weak social connections prevent a network of mobile phone calls from fragmenting into a large number of disconnected components [13], and are thus critical avenues for information to diffuse widely across the network.

## 11.7 Concluding Remarks

This chapter presented a collection of recently developed propagation models used for mobile phone data analytics. This collection of models revealed important statistical qualities of information propagation processes over mobile phone networks, were used to model unique propagation phenomena, and utilized in a number of novel applications. Based on the qualities of the models, it identified a number of open opportunities for researchers to develop ever more sophisticated and realistic models of propagation phenomenon within mobile phone networks.

## References

1. 800notes: Directory of unknown callers. http://www.whocallsme.com
2. Almeida, T.A., Hidalgo, J.M.G., Yamakami, A.: Contributions to the study of sms spam filtering: new collection and results. In: Proceedings of 11th ACM Symposium on Document Engineering, pp. 259–262. ACM (2011)
3. Anderson, R.M., May, R.M., Anderson, B.: Infectious diseases of humans: dynamics and control, vol. 28. Wiley Online Library (1992)
4. Barron, A., Rissanen, J., Yu, B.: The minimum description length principle in coding and modeling. IEEE Trans. Inf. Theory **44**(6), 2743–2760 (1998)
5. Berlingerio, M., Calabrese, F., Di Lorenzo, G., Nair, R., Pinelli, F., Sbodio, M.L.: Allaboard: a system for exploring urban mobility and optimizing public transport using cellphone data. In: Machine Learning and Knowledge Discovery in Databases, pp. 663–666. Springer (2013)
6. Bjelland, J., Canright, G., Engo-Monsen, K., Sundsoy, P.R., Ling, R.S.: A social network study of the apple vs. android smartphone battle. In: Proceedings of International Conference on Advances in Social Networks Analysis and Mining, pp. 983–987. IEEE Computer Society (2012)
7. Candia, J., González, M.C., Wang, P., Schoenharl, T., Madey, G., Barabási, A.L.: Uncovering individual and collective human dynamics from mobile phone records. J. Phys. A: Math. Theor. **41**, 11 pp (2008)
8. Chen, W., Wang, Y., Yang, S.: Efficient influence maximization in social networks. In: Proceedings of 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 199–208. ACM (2009)
9. Chien, E.: Security response: Symbos. mabir. Techical report. Symantec Corporation (2005)
10. Corporation, I.: Global business security index report. Technical report. IBM (2004)
11. Dasgupta, K., Singh, R., Viswanathan, B., Chakraborty, D., Mukherjea, S., Nanavati, A.: Social ties and their relevance to churn in mobile telecom networks. In: Proceedings of 11th ACM International Conference on Extending Database Technology (2008)
12. Doran, D., Alhazmi, H., Gokhale, S.: Triads, transitivity, and social effects in user interactions on facebook. In: Proceedings of IEEE International Conference on Computational Aspects of Social Networks, pp. 68–73 (2013)
13. Doran, D., Mendiratta, V., Phadke, C., Uzunalioglu, H.: The importance of outlier relationships in mobile call graphs. In: Proceedings of International Conference on Machine Learning and Applications, pp. 24–29 (2012)
14. Eagle, N., Pentland, A.S., Lazer, D.: Inferring friendship network structure by using mobile phone data. Proc. Natl. Acad. Sci. **106**(36), 15274–15278 (2009)
15. Ferrie, P., Szor, P., Stanev, R., Mouritzen, R.: Security response: SymbOS. Symantic Corporation, Cabir. Technical repot (2004)
16. Fessenden-Raden, J., Fitchen, J.M., Heath, J.S.: Providing risk information in communities: Factors influencing what is heard and accepted. Sci. Technol. Hum. Values **12**, 94–101 (1987)
17. Gleave, E., Welser, H.T., Lento, T.M., Smith, M.A.: A conceptual and operational definition of 'social role' in online community. In: 42nd Hawaii International Conference on System Sciences, pp. 1–11 (2009)
18. Groenevelt, R., Nain, P., Koole, G.: The message delay in mobile ad hoc networks. Perform. Eval. **62**(1), 210–228 (2005)
19. Jiang, N., Jin, Y., Skudlark, A., Hsu, W.L., Jacobson, G., Prakasam, S., Zhang, Z.L.: Isolating and analyzing fraud activities in a large cellular network via voice call graph analysis. In: Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services, pp. 253–266. ACM (2012)
20. Kaspersky: Kaspersky security bulletin malware evolution. Kaspersky Security Bulletin Malware Evolution 2011
21. Kempe, D., Kleinberg, J., Tardos, E.: Maximizing the spread of influence through a social network. In: Proceedings of 9th ACM International Conference on Knowledge Discovery and Data Mining, pp. 137–146 (2003)

22. Kephart, J.O., White, S.R.: Directed-graph epidemiological models of computer viruses. In: Proceedings of IEEE Computer Society Symposium on Research in Security and Privacy, pp. 343–359. IEEE (1991)
23. Kermack, W.O., McKendrick, A.G.: Contributions to the mathematical theory of epidemics. part i. Proc. Roy. Soc. Lond. Ser. A **115**(5), 700–721 (1927)
24. Kermack, W.O., McKendrick, A.G.: Contributions to the mathematical theory of epidemics. ii. the problem of endemicity. Proc. Roy. Soc. Lond. Ser. A **138**(834), 55–83 (1932)
25. Khouzani, M., Sarkar, S., Altman, E.: Maximum damage malware attack in mobile wireless networks. IEEE/ACM Trans. Netw. **20**(5), 1347–1360 (2012)
26. Kim, H., Zang, H., Ma, X.: Analyzing and modeling temporal patterns of human contacts in cellular networks. In: Proceedings of 22nd IEEE International Conference on Computer Communications and Networks, pp. 1–7 (2013)
27. Kitagawa, G., Gersch, W.: Smoothness Priors Analysis of Time Series, vol. 116. Springer (1996)
28. Krings, G., Calabrese, F., Ratti, C., Blondel, V.D.: Urban gravity: a model for inter-city telecommunication flows. J. Stat. Mech.: Theory Exp. L07003 (2009)
29. Król, D.: Propagation phenomenon in complex networks: theory and practice. New Gener. Comput. **32**(3–4), 187–192 (2014)
30. Kurtz, T.G.: Solutions of ordinary differential equations as limits of pure jump markov processes. J. Appl. Probab. **7**(1), 49–58 (1970)
31. Lambiotte, R., Blondel, V.D., de Kerchove, C., Huens, E., Prieur, C., Smoreda, Z., Van Dooren, P.: Geographical dispersal of mobile communication networks. Phys. A Stat. Mech. Appl. **387**(21), 5317–5325 (2008)
32. Leskovec, J., Krause, A., Guestrin, C., Faloutsos, C., VanBriesen, J., Glance, N.: Cost-effective outbreak detection in networks. In: Proceedings of 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 420–429. ACM (2007)
33. Liben-Nowell, D., Novak, J., Kumar, R., Raghavan, P., Tomkins, A.: Geographic routing in social networks. Proc. Natl. Acad. Sci. USA **102**(33), 11623–11628 (2005)
34. Mao, H., Shuai, X., Ahn, Y.Y., Bollen, J.: Mobile communications reveal the regional economy in cote d'ivoire. In: Proceedings of International Conference on Analysis of Mobile Phone Datasets and Networks D4D Book, pp. 1–18 (2013)
35. Mickens, J.W., Noble, B.D.: Modeling epidemic spreading in mobile environments. In: Proceedings of the 4th ACM Workshop on Wireless Security, pp. 77–86. ACM (2005)
36. Montoliu, R., Gatica-Perez, D.: Discovering human places of interest from multimodal mobile phone data. In: Proceedings of 9th International Conference on Mobile and Ubiquitous Multimedia, pp. 12:1–12:10. ACM (2010)
37. Pan, W., Aharony, N., Pentland, A.: Composite social network for predicting mobile apps installation. In: Proceedings of AAAI Conference on Artificial Intelligence, pp. 821–827 (2011)
38. Peng, S., Yu, S., Yang, A.: Smartphone malware and its propagation modeling: a survey. IEEE Commun. Surv. Tutor. **16**(2), 925–941 (2014)
39. Peruani, F., Tabourier, L.: Directedness of information flow in mobile phone communication networks. PloS One **6**(12), e28,860 (2011)
40. Phadke, C., Mendiratta, V., Uzunalioglu, H., Doran, D.: Prediction of subscriber churn using social network analysis. Bell Labs Tech. J. **17**(4), 63–75 (2013)
41. Phithakkitnukoon, S., Horanont, T., Di Lorenzo, G., Shibasaki, R., Ratti, C.: Activity-aware map: identifying human daily activity pattern using mobile phone data. In: Human Behavior Understanding, pp. 14–25. Springer, Berlin (2010)
42. Rhodes, C.J., Nekovee, M.: The opportunistic transmission of wireless worms between mobile devices. Phys. A Stat. Mech. Appl. **387**(27), 6837–6844 (2008)
43. Rivera, M.T., Soderstrom, S.B., Uzzi, B.: Dynamics of dyads in social networks: assortative, relational, and proximity mechanisms. Annu. Rev. Sociol. **36**, 91–115 (2010)
44. Sellnow, T.L., Ziegelmueller, G.: The persuasive speaking contest: an analysis of twenty years of change. Natl. Forensic J. **6**(2), 75–87 (1988)
45. Systems, JJuniper: Mobile Threats Report. In: Technical report (2011)

46. Taillard, M.O.: Persuasive communication: the case of marketing. Working Papers in Linguistics, vol. 12, pp. 145–174 (2000)
47. Trivedi, K.S.: Probability and Statistics with Reliability, Queueing, and Computer Science Applications, 2nd edn. Wiley, New York (2002)
48. Wang, D., Pedreschi, D., Song, C., Giannotti, F., Barabasi, A.L.: Human mobility, social ties, and link prediction. In: Proceedings of 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1100–1108. ACM (2011)
49. Wang, Y., Cong, G., Song, G., Xie, K.: Community-based greedy algorithm for mining top-k influential nodes in mobile social networks. In: Proceedings of 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1039–1048. ACM (2010)
50. WhoCallsMe: Reverse phone number lookup. http://www.whocallsme.com
51. Zhang, W., Li, Z., Hu, Y., Xia, W.: Cluster features of bluetooth mobile phone virus and research on strategies of control and prevention. In: Proceedings of International Conference on Computational Intelligence and Security, pp. 474–477. IEEE (2010)

# Chapter 12
# Information Propagation in Social Networks During Crises: A Structural Framework

**Daniela Pohl and Abdelhamid Bouchachia**

**Abstract** In crisis situations like riots, earthquakes, storms, etc. information plays a central role in the process of organizing interventions and decision making. Due to their increasing use during crises, social media (SM) represents a valuable source of information that could help obtain a full picture of people needs and concerns. In this chapter, we highlight the importance of SM networks in crisis management (CM) to show how information is propagated through. The chapter also summarizes the current state of research related to information propagation in SM networks during crises. In particular three classes of information propagation research categories are identified: network analysis and community detection, role and topic-oriented information propagation, and infrastructure-oriented information propagation. The chapter describes an analysis framework that deals with structural information propagation for crisis management purposes. Structural propagation is about broadcasting specific information obtained from social media networks to targeted sinks/receivers/hubs like emergency agencies, police department, fire department, etc. Specifically, the framework aims to identify the discussion topics, known as sub-events, related to a crisis (event) from SM contents. A brief description of techniques used to detect topics and the way those topics can be used in structural information propagation are presented.

## 12.1 Introduction

The last decades brought several technical innovations which changed peoples' communication behavior due to light-weight, powerful mobile phones, mobile Internet, and mobile applications. With these innovations in recent years, social media

D. Pohl
Institute of Information Technology, Alpen-Adria-Universität Klagenfurt,
Universit ätsstr. 65-67, 9020 Klagenfurt, Austria
e-mail: daniela@itec.aau.at

A. Bouchachia (✉)
Smart Technology Research Center, Bournemouth University,
Fern Barrow Poole BH12 5BB, UK
e-mail: abouchachia@bournemouth.ac.uk

D. Król et al. (eds.), *Propagation Phenomena in Real World Networks*,
Intelligent Systems Reference Library 85, DOI 10.1007/978-3-319-15916-4_12

networks (e.g., Twitter, Flickr, YouTube, Facebook etc.) form new communication and news channels. People have now the possibility to document every situation they are involved in. They can propagate updates, requests, opinions, and information of general public interest.

In other terms, social media users are actually producer, repeater and consumer of information—all at the same time. SM networks facilitate the dissemination of information along social links connecting communities formed in these networks. These connections are of two types: strong ties which are usually formed by frequent contact (family and friends) and weak ties which are formed by infrequent contact.

Over the past SM networks have been studied by different communities focussing on various aspects related to the identification of communities, the propagation of information through the network, modeling the relationship between information producer and information consumer, analysis of sentiment, and topic detection. Due to these advances, SM have emerged as research avenue in the area of crisis management studies. As stated in Sect. 12.2 below, several studies highlight the importance of social media in crisis management. People tend to use SM particularly when communicating with emergency services is difficult due to overload or emergency services could not be on site sufficiently early when the crisis happened.

SM networks can be analyzed to uncover topics discussed during a crisis, to identify the people needs, and to discover rumor so that it can systematically be countered. Therefore, efficient methods for analyzing SM need to be developed in order to:

- detect the important information related to people needs, human casualties and infrastructure damages from SM contents,
- understand the community and network structure, and
- find communication hubs to propagate useful information that can efficiently be exploited to stabilize the situation, give hints, publish shelter information, and counter against rumors.

This chapter summarizes the research on information propagation in SM networks for crisis management. A framework for dealing with structural information propagation is described. Methods for analysis of SM content, developed by the authors,[1] are discussed to show how *structural information propagation* can be used in crisis management.

The rest of the chapter is organized as follows. Section 12.2 highlights the relationship between social media and crisis management by showing illustrative recent studies. Section 12.3 summarizes research that focuses on information propagation in social media during a crisis. Section 12.4 proposes an analysis framework for disseminating and propagating valuable information to be used by targeted people, like first responders, in order to organize their intervention. In particular, the framework describes how topics discussed in SM networks during crises are detected and tracked to be propagated later on to targeted people. Section 12.5 concludes the chapter.

[1] We don't claim to provide any original results in this chapter. We simply summarize the different research studies relevant to information propagation in the context of social media and crisis management.

## 12.2 Crisis Management and Social Media

Crisis Management (CM) referring to disaster, emergency or catastrophe management indicate all actions taken before, during, and after a disaster [14]. In general, CM tasks are divided into four phases [14]: *mitigation*, *preparedness*, *response* and *recovery*.

- *Mitigation* deals with challenges regarding risk assessment and crisis prevention to minimize the possibility of disaster occurrence.
- *Preparedness* includes all steps to increase readiness in case of disaster (e.g., training, public awareness, etc.).
- *Response* covers actions to reduce causalities and to stabilize the situation during the disaster.
- *Recovery* focuses on re-establishing the damaged infrastructure and facilitating normal course of life.

In all these phases, information about the situation at hand is important for situational awareness, planning, and decision making.

Research shows that SM networks are increasingly used during crisis situations, e.g., to publish information on the current situation. There are several SM studies highlighting the importance of different SM networks. For instance, Palen [25] described the usage of SM platforms during the Virginia Tech shooting and the southern California wildfire. Vieweg et al. [47] showed how Twitter was used during two emergencies: Red river floods and the Oklahoma grassfire in 2009. They analyzed tweets and identified different categories of tweets (i.e., warning, information on impact, weather, evacuation, etc.). Choudhary et al. [9] studied tweets obtained during the Egyptian uprising. In particular, they analyzed the sentiment and the topics discussed during the event. Reuters et al. [39] analyzed SM in the context of emergent groups during a disaster. They identified several roles of users in such groups, e.g., reporters who generate contents, retweeter who forward and propagate the information, etc. They also studied SM in the context of European incidents, e.g., the eruption of Eyjafjallajkull and the mass-panic during Love parade in Duisburg [40]. Terpstra et al. [45] analyzed the data collected during a festival in Belgium (Pukkelpop) which was hit by a heavy storm. They used keyword-based filters to identify tweets with specific content, like damages or causalities. Their analysis showed that tweets can be seen as realtime information sources for situational awareness in crisis management. Perng et al. [26] analyzed Twitter in the context of the Norway attacks in 2011.

Beside Twitter and Facebook, photo sharing tools, like Flickr, have been used to exchange information about different events/incidents. Liu et al. [20] described the importance of photo sharing via SM during emergency cases. The work depicted the usage of Flickr through six different crisis between 2004 and 2007, e.g., Indian Ocean earthquake, Hurricane Katrina, Minneapolis bridge collapse, etc. They showed that there were eyewitness photos which can be used for a formal emergency response [20]. Flickr was analyzed by Fontugne et al. [13] in the context of the Japan Tsunami. The study showed that it is possible to identify crisis situations and their impact using the content of Flickr, in particular the metadata of the crisis pictures. Dashti et al. [10] studied Twitter data for disaster recovery and reconnaissance. They argued that the most important messages are those including pictures and location information. Yates and Paquette [50] described the use of SM during Haiti earthquake. They highlighted the importance of SM for communication and collaboration between aid units to share information in an easy and efficient way. Likewise, Dugdale et al. [12] analyzed the use of SM during Haiti earthquake. The results show that SM is important for situational awareness stating the positive attitude of first responders to use social networks. Hughes et al. [16] collected data related to Hurricane Sandy from Twitter, Facebook, agencies websites and Nixle which is a notification service. They studied online communication of local fire and police services. The study showed that efficient browsing techniques for analysis of SM data are important.

Agencies make use of social networks to monitor the reaction of the public on different events and to communicate with the public during such events. For example, Denef et al. [11] analyzed the tweeting behavior of two major police forces in UK during the 2011 riots. They highlighted the different strategies of the police forces in communicating with the public. They also presented the different pro and cons of these strategies. Although, relief units use SM in CM and daily work, development of new practice and supporting technologies (e.g., identify active users, understand the current information propagation trend, detect topics and events, etc.) are needed to connect both communities, the public and emergency services [7].

## 12.3 Information Propagation in Crisis Situations

SM networks offer simple communication channels to propagate information in a short time. Many research studies illustrate how information can be disseminated in realtime which enable just-in-time analysis of SM data upon the occurrence of events. For example, Stollberg and de Groeve [43] showed that it is possible to detect events by monitoring tweets that contain specific keywords (e.g., earthquake).

In the following we address important research topics in relation to SM and CM:

- *Network analysis and community detection*: This is about the analysis of SM networks in static or dynamic fashion such as topology and changes of the network. The aim is to detect communities and uncover the communication channels for information spreading. It facilitates the analysis of ties within the communities. Details follow in Sect. 12.3.1.
- *Role and topic-oriented information propagation*: Analysis of which kind of information is propagated through SM networks and which users will receive information. The question about which information to trust is another area that falls into this category. Details follow in Sect. 12.3.2.

- *Infrastructure-oriented information propagation*: It is about structural forwarding of information to responsible persons or facilitating an infrastructure to forward information. Details follow in Sect. 12.3.3.
- *Prediction and forecasting*: This is related to forecasting of several aspects like evolution, impact, behavior of users, etc. Usually the reliability of forecasting depends on the characteristics of the propagation itself [8]. The present chapter will not further discuss this topic as it is out of the intended scope.

The first two points are related to each other, because if the structure of the network is known, the mechanisms/ways the information propagates can also be modeled.

Information published on SM networks can emanate from two sources:

- *General public*: un-structural information can take different forms: text, images, videos.
- *Crisis management*: structural information exchange and propagation within CM services.

Most of the research studies investigate data emanating from the public in order to enhance situation awareness, decision making, and resource management.

### *12.3.1 Network Analysis and Community-Detection*

Tyshchuk et al. [46] examined Twitter data related to Japan tsunami in 2011 by considering social network analysis and natural language processing to understand and uncover communication patterns. The social network analysis was used to find and analyze communities using a random walk algorithm on time slices. Twitter texts were investigated to understand the role of community members [46].

Ren et al. [38] described a visual analytic tool called WeiboEvents.[2] It comprises two parts: (1) a web-based analysis tool and (2) an expert analytic system. The tools are used to analyze and explore a retweet tree and to identify clusters of users. To get rid of the huge amount of data in SM networks, the web-based tool is used as crowd-sourcing tool, where the user can solve mini-tasks asked by the system (i.e., annotating tweets or the importance of users). The results are stored and shared with other users, allowing also an expert in front of the expert analytic system (in case of an emergency) to reuse already examined data. The system is used in China since 2012.

Kumar et al. [19] and Morstatter et al. [22] described TweetXplorer, a tool to identify: who is important, where do relevant tweets originate from, and when do different trends occur. The number of retweets is used to identify important users and tweets. Additionally, time and location are used in visual analytics. TweetXplorer visualizes the obtained retweet-network in a graph, map and keyword-cloud representation. The graph represents the retweet network, the map shows the heat-map of tweets and the most important keywords of a time period are visualized in a word cloud. They evaluated their tool using data about Hurricane Sandy. The experiments

[2] Weibo is a very popular Chinese microblogging platform.

showed that clusters of users discussing specific topics (e.g., power outage, hospital) can be identified [22].

Sutton et al. [44] examined the amplification of tweets during Boston bombing in 2013. They collected data originating from organizations responsible for emergency management during the incident from Twitter. The data was used to identify users or other organization (i.e., user profiles) in the network which amplify information by retweeting the original tweet [44]. These amplifying user profiles must be considered in a standardized communication process (helping to support data propagation). To identify such users, they analyzed the tweets through creating a retweet-network, where the nodes represent the different organizations and the edges show interaction (i.e., if one node retweets messages from the other node) [44]. The influence of a node is given by the amount of retweets the node performed. It shows that local organization has the most influence in creating new tweets/information and non-local organization acts as supporter for the propagation of the information through retweeting.

Klein et al. [18] suggested a framework based on graph analytic to analyze the Twitter network during an incident in real-time. They tested a framework based on tweets gathered from Twitter with keywords related to emergencies (e.g., earthquake, flood, etc.). It tracks the communication network behind the tweets to identify the leading members in the network. In addition, the content of tweets itself is also analyzed through lexical analysis.

Most of the works focus on Twitter, due to the easy access and the fact that Twitter is a platform with real-time characteristic during a crisis [41]. The presented research work above showed the identification and analysis of the networks based on activities/retweets between users.

Network analysis and community detection are important for CM, especially in the phases *preparedness* and *response*, because they help (1) to understand how information is propagated, (2) to identify sources of rumors and (3) to create more efficient communication plans.

### *12.3.2 Role and Topic-Oriented Information Propagation*

Starbird and Palen [42] analyzed the spread of information, especially, the retweet behavior during the Egyptian uprising. They collected a huge number of tweets, approximately 2.2 million tweets. 956 most tweeting user accounts where analyzed and labeled manually using the following coding: *in Cairo*, *in Egypt but not in Cairo*, and *not in Egypt*. Tweets that are propagated are related to solidarity, detained friends/relatives and violence. The most retweeted information was sent from local accounts in Cairo, for example, from journalists and bloggers. The authors also stated that the retweeting behavior of other users served as “crowd-based approval” of the content propagated.

Purohit et al. [36] showed that it is possible to identify influential users (i.e., virtual responders). These users can be regarded as disseminators to communicate information to the general public. The identification of influential users is based on a graph built from tweets lying in a specific time window. Nodes represent users and directed edges represent the interaction between users. Additionally, the profession of users can be identified by comparing the user profile with a lexicon representing the most important roles (i.e., journalism, police, etc.) during an emergency. Visualization was used to browse through the graph and identify propagation paths of false information.

Mendoza et al. [21] analyzed tweets from the Chilean earthquake in 2010. They examined the correlation between followers and number of tweets. The correlation showed that the most active users have a high amount of followers in the network. The topology of the network also uncovers related communities. The authors examined the propagation of tweets containing the word "earthquake". They found different patterns of propagation (i.e., tree-based and cyclic). Tree-based patterns produce direct information propagation, while cyclic patterns represent comments or replies. Very interesting was the difference in propagation of rumors and true information. Rumors tended to cause more questions by other persons in the network and therefore it is possible to detect rumors easily.

Ireson et al. [17] analyzed local communities in the case of Sheffield flooding in 2007. They used local forums and blog posts to analyze the frequency of posts from individuals. They analyzed groups related to topics. It turned out that users with high-frequency posts provided important information to the situation at hand. Reuter et al. [39, 40] analyzed incidents in Europe and the USA. They distinguished between virtual and real emergent groups. They identified several user groups by roles [39]: *helper*, *repeater*, *retweeter*, and *reporter*. Hughes and Palen [15] identified Twitter as a major broadcast medium for information dissemination. They showed that some users act as "disseminators" in collecting and disseminating information. In [16], Hughes et al. also analyzed SM from another perspective: how is SM used by police &fire services to propagate information to the public. They examined data collected from Facebook, Twitter, Nixle and the website of the police/firefighters department related to the Hurricane Sandy in 2012. The results showed that the most important communication channel was Facebook followed by Twitter. They can be used to communicate closure of transportation or areas and to publish information from third parties like weather conditions.

The identification of roles allows to refine communication plans for disseminating information in a fast way, for example during *response*.

### *12.3.3 Infrastructure-Oriented Information Propagation*

Propagation of information is not always straightforward, especially when the communication infrastructure has (partially) been destroyed. This is very important before or during *recovery* phase of CM. Therefore, research focuses on ad-hoc networks for forwarding messages (i.e., tweets). Al-Akkad et al. [1] discussed the physical perspective of information spreading focusing on a store-and-forward mechanism to overcome the limitation of the damaged infrastructure. They developed an application for Android mobile phones which takes care of the publication of tweets. It uses WiFi technology to join and leave “islands of connectivity” [40]. If a mobile device reaches an island with Internet access (i.e., another mobile phone in the ad-hoc network with Internet connection), messages stored on the mobile phone get published. Similar systems can be found in [1]. Moreover, there are also different routing algorithms to transmit information within ad-hoc networks in an efficient way. Raffelsberger and Hellwagner [37] described a comparison of routing algorithms in context of emergency response. Zhou et al. [51] collected microblogs from Weibo to build a Naïve Bayes classifier that allows routing information to emergency departments (e.g., police, fire department, ambulances, etc.).

Routing of information was also examined in the context of first responders (i.e., not only for social media data). In Netten et al. [23, 24], for example, important information is forwarded to people also interested in that information or topic due to their role. They recorded conversion between relief units which was then processed to detect conversation topics.

Wei et al. [48] discussed the difference in disseminating information through strong and weak ties during disaster situations. They created and ran a simulation environment emulating these strong and weak ties. The results showed that weak ties have a high influence in propagating information, as they often have a bridging function between communities with strong ties. This has also been shown by Bakshy et al. [3].

Infrastructure-oriented propagation is important particularly during the *response* and *recovery* phases when the infrastructure is not completely restored. Efficient and intelligent routing mechanism of information helps propagating important facts about the crisis.

## 12.4 From SM to Structural Information Propagation

In crisis management information must be first uncovered, analyzed and then directed to people in particular *structures* such as emergency services who will make decision. In the following we describe a framework for structural information propagation during emergency management. Figure 12.1 sketches the general processing chain for analyzing SM in CM. First, in the *information propagation* step, the people (e.g., including victims, involved people, bystanders, etc.) propagate information about the current situation.

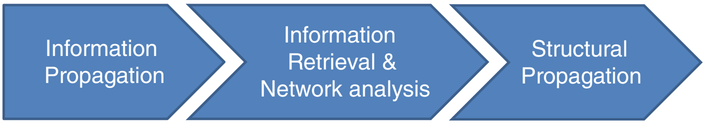


**Fig. 12.1** General processing chain

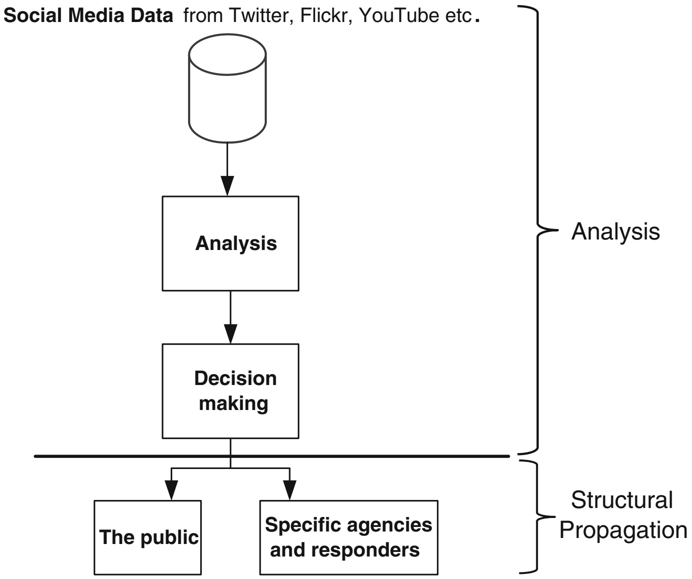

Social Media Data from Twitter, Flickr, YouTube etc.
Analysis
Decision making
The public
Specific agencies and responders
Analysis
Structural Propagation

**Fig. 12.2** Information processing and propagation: detailed framework

The information is analyzed using various techniques stemming from *Information Retrieval* and *Network Analysis* by examining the network structure, role and discussion topics. In [30], we investigated social media analysis approach to detect topics related to a crisis. The outcome of the investigations are then used for *structural propagation*, by redirecting the topics to the crisis services.

Figure 12.2 provides details of the last two steps. Relying on specific interfaces such as the Twitter Streaming API, relevant data is obtained. Results of the *analysis* are then used for decision making. Information can also be published for the *public* and redirected to the *first responders* or other *agencies* such as police and fire departments.

Information propagated through the network may call for some correcting actions. The reason for corrective actions could be rumors or false information, but also additional information for prevention. Social media can be used to convey information to the public. Results of the network analysis can be used to propagate information in a focused and efficient way.

To track structural information, it is important to understand which topic is currently being discussed on social networks, to interpret that topic, and to develop the appropriate action strategy based on what is currently discussed and propagated.

We have developed several topic detection algorithms for social media. Topics can be hotspots or incidents like flooding, collapsing of bridges, etc. in a specific area. These topics are called *sub-events* which can be identified by accumulating/aggregating all messages posted in relation to the same topic. Retweets/re-posts amplify the importance of topics as the amount of messages discussing the

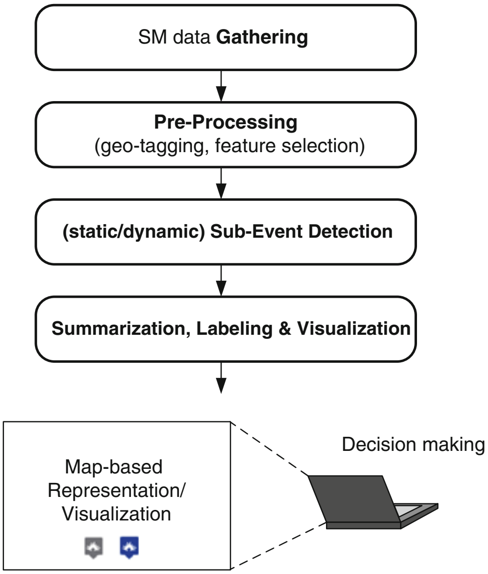

SM data **Gathering**
**Pre-Processing**
(geo-tagging, feature selection)
**(static/dynamic) Sub-Event Detection**
**Summarization, Labeling & Visualization**
Map-based Representation/ Visualization
Decision making

**Fig. 12.3** Topic detection as analytic instrument for structural propagation (Markers by MapIcon-Collection mapicons.nicolasmollet.com)

topic increases. The advantages of the framework are (1) to uncover the important sub-events during a crisis, and (2) to propagate the analysis results to agencies or other involved parties. Our work on detecting sub-events from SM data is summarized in the following section.

### *12.4.1 Topic Detection*

To uncover the discussion topics in SM during a crisis, we developed clustering approaches. Two strategies were investigated: offline where a corpus is readily available and online where data is obtained and sequentially processed in realtime.

Offline clustering methods were used to detect sub-events which help understand how and what people communicated about an event. Moreover, the offline strategy is used aftermath and for training purposes by the agencies. In parallel we developed an online method which was integrated into a system tested during an exercise in the BRIDGE project.[3]

The different processing steps for the offline and online approaches are summarized in Fig. 12.3. First, data is gathered via standardized APIs from the SM networks. Data is then geo-tagged before text analysis is performed. Important terms/words are selected through appropriate features selection methods. Afterwards, the offline or online (topic) sub-event detection processes are executed

[3] http://www.bridgeproject.eu/en/news/BRIDGE-Conducts-Its-Third-Successful-Demonstration-in-Stavanger--Norway_126, (Accessed: July 2014).

(see Sects. 12.4.3 and 12.4.4). The resulting topics are summarized, labeled with human-readable keywords and visualized in a user-friendly interface.

Using the visualization tool, a person (or team) decides what to do. The following scenarios can be envisaged:

- They could be ignored if the identified issues are already handled by first responders or are out-dated.
- The outcome is transmitted to specific persons on-site to take actions (response, rescue).
- Steps must be taken to overcome rumors or clarify the situation in the public.

The propagation could be supported by results of network analysis to find information hubs to disseminate and spread information in a more efficient way.

### *12.4.2 Datasets*

In our studies we used different datasets as shown in Table 12.1. The datasets were collected from several social media networks (i.e., YouTube, Flickr and Twitter) based on indicators related to the emergency situation at hand. Search indicators are usually keywords that reflect on the targeted situation. The datasets are related to crises of different magnitudes. For example, the Mississippi Flood crisis consist of many sub-events describing flood in cities, like Vicksburg and New Orleans. The same is true for the Hurricane Irene which affected South Carolina, Virginia and New York. As to UK riots, different cities are affected like London, Manchester and Birmingham.

The first four datasets in Table 12.1 were used in studies for offline clustering. For the evaluation, the UK riots dataset was labeled based on the sub-events that could be identified by human using textual features and metadata [31]. Labeling requires the location, time and content to identify sub-events (see Sect. 12.4.3).

Hurricane Sandy was used to analyze the online clustering strategy (see Table 12.1). A large amount of media items for a shorter time period has been used to track the dynamics of social media networks. We used Twitter as real-time broadcast medium to collect relevant messages.

In the following sections, we present a summary of examined offline and online approaches.

**Table 12.1** Datasets related to different disasters [33]

| Dataset name | Duration | #Pictures | #Videos | #Tweets | Σ items |
|---|---|---|---|---|---|
| Mississippi flood (MF) | 04–19 May 2011 | 2,039 | 442 | 0 | 2,481 |
| Oslo bombing (OB) | 22 July 2011 | 31 | 222 | 0 | 253 |
| UK riots (UK) | 06–10 Aug 2011 | 178 | 274 | 0 | 452 |
| Hurricane Irene (HI) | 23–29 Aug 2011 | 455 | 700 | 0 | 1,155 |
| Hurricane Sandy (HS) | 29–30 October 2012 | 286 | 167 | 1,003 | 1,456 |

### 12.4.3 *Offline Approaches*

In the offline clustering study, we have proposed several methods for sub-event detection:

- *Self-organizing maps* (*SOM*) in [28] We used the MF, OB, UK, and HI datasets shown in Table 12.1. We compared our results with the ground truth (i.e., describing the timeline of the different incidents: MF: [34], OB: [5], UK: [4], and HI: [2, 35]).
- *Agglomerative clustering* (*AC*) in [29] Agglomerative clustering is a clustering approach which allows to visualize intermediate result-steps for the user in an intuitive representation, called dendrogram. We introduced a weighting mechanism for keywords/features based on the location of their appearance (title, body, tags). We also used the MF, OB, UK, and HI datasets and compared them to the ground truth.
- *2 Phase-Geo* (*2PG*) in [27] We also introduced a 2 Phase-Geo (2PG) clustering approach based on self-organizing maps. It allows to take the (often) sparse geo-information into account when performing the clustering.
- *2 Phase-Geo-Time* (*2PGT*) in [27] In another variation of the 2PG algorithm, we additionally considered the time-information of media items published. We evaluated both algorithms, 2PG and 2PGT, using MF, OB, UK and HI datasets to show the impact of geo-information.

Figure 12.3 summarizes the steps performed during the analysis. The geo-tagging step is used only for approaches which need that tag in order to run the detection procedure.

In [31] we compared the SOM, AC, 2PG, and 2PGT algorithms using different criteria: *scalability, metadata quality, ground truth* and *clustering quality*. *Scalability* compares the runtime complexity, the needed parameters and the visualization of the clustering results for their clarity. *Metadata quality* examines the sensitivity of the clustering approaches regarding the completeness of the metadata. We compared the clustering results against the corresponding ground truth.

The clustering quality is measured based on the *topic* and *item level*. For the *topic level* we check if the different approaches were able to identify the topics. We made use of clustering quality metrics (Dunn, Davies-Bouldin and Silhouette Index [31]) and the Normalized Mutual Information (NMI) to compare the similarity between the clustering results [31]. 2PG and 2PGT show an intuitive interface to visualize the data (i.e., map-based representation) compared to the other algorithms focusing on text-based visualization.

In addition, we compared different clustering approaches based on *item-level*. We labeled UK riots dataset with different granularity. First, labeling was performed with the focus on the City-District-Incident-Date (CDID) format. This means, items in the dataset with the same city, district, and describing the same incident (on the same date) are labeled similarly. We considered also a City-District (i.e., topics from the same city and district are summarized) and City level (i.e., topics related to the same city are combined). For comparison purposes, we used the purity metric to

**Table 12.2** Short summary of the evaluation, ($n \hat{=}$ number of media items) [31]

| | SOM | AC | 2PG | 2PGT |
|---|---|---|---|---|
| Performance | $O(n^2)$ | $O(n^3)$ | $O(n^2)$ | |
| Parameter | Number of clusters and terms used in the aggregation | | | |
| Representation | Table-representation with text rows | | Map-based representation | |
| Metadata | Text | | Text+Geo | Text+Geo +Time |
| Ground truth (topic-level) | Identifies most important sub-events<br>2PG and 2PGT easier to read | | | |
| Cluster quality (topic-level) | Similar in assigning data items | | | Future extension with streaming |
| Cluster quality (item-level) | AC and 2PGT performs best on item level evaluation | | | |

identify the best algorithm and we showed that AC and the 2PGT perform best for the different granularity levels.

The results are summarized in Table 12.2 based on the different criteria. For further details please refer to [31] and [33].

It is easy to communicate those sub-events that are the focus of discussion in SM networks to the interested parties (i.e., relief units, incident commander, etc.). In an aftermath analysis, the obtained insights can be used to simulate or model the propagation of information for an exercise.

### *12.4.4 Online Approaches*

Based on the findings of the offline approaches, we developed an online approach to detect and track sub-events in real-time during a crisis. We developed an online feature selection mechanism combined with an online clustering algorithm [30]. The feature selection approach extends the term-frequency-inverse document frequency (tf-idf) by introducing an additional weighting mechanism for trending keywords. This means, keywords with a high peak in incoming messages are highly weighted than keywords with less frequency. So, the most important features are identified for further consideration in the clustering stage.

Growing Gaussian Mixture Models [6] have been used to cluster the items with the feature sets resulting form the processing steps. Each cluster is represented as a multivariate Gaussian. The algorithm allows to split and merge existing clusters, to optimize the number of clusters. In the best case, each cluster represents one sub-event (topic). Outdated features are removed from the clusters by deleting the corresponding entries from the multivariate Gaussian’ center and covariance matrix.

We evaluated the clustering algorithm using the Hurricane Sandy (HS) data [30] (see Table 12.1). Compared to the ground truth (HS:[49]), the algorithm identifies topics related to flood, power outage and different damages. Additionally, we compared the algorithm against a baseline offline algorithm, which calculates the features in advance from the entire dataset. Using Normalized Mutual Information (NMI) the online clustering showed a similar behavior as the baseline algorithm, although features are dynamically examined. We used the Silhouette metric to judge the quality of the clusterings: high Silhouette values indicate well separated clusters. The values for the online and offline approach are very similar (e.g., in average a difference of 0.16) [30].

We incorporated the developed online features selection and clustering approach in an real-time SM analysis tool called "Information Aggregator"[4] for supporting crisis management. The tool allows the transmission of important outcome to first responders on-site of the incident [33] as a typical case of structural information propagation. That is, identified sub-events are used during crisis management as additional source for decision making and for defining new communication strategies. First responders see a benefit of using knowledge gained from SM to execute management tasks [32].

## 12.5 Conclusion and Future Work

This chapter highlights the importance of social media in crisis management. It describes research on how information is propagated through social networks during crises. We categorized information propagation into three classes: network analysis and community detection, role and topic-oriented information propagation, infrastructure-oriented information propagation. We suggested a framework for structural information propagation, that is about how hot topics discussed by people can be detected and forwarded to first responders or used to communicate information to the public. We briefly summarized our work in relation to topic detection using various offline and online algorithms.

In future, we plan to extend the topic detection framework by considering active learning to enhance the quality and efficiency of topic detection. We also plan to use the network structure to enable efficient information propagation.

**Acknowledgments** The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement no 261817 and was partly performed in the Lakeside Labs research cluster at Alpen-Adria-Universität Klagenfurt.

[4] http://www.bridgeproject.eu/content/bridge_information_intelligence_flyer.pdf, (Accessed: July 2014).

## References

1. Al-Akkad, A., Raffelsberger, C., Boden, A., Ramirez, L., Zimmermann, A.: Tweeting 'when online is off'? opportunistically creating mobile ad-hoc networks in response to disrupted infrastructure. In: Proceedings of the 11th International Conference on Information Systems for Crisis Response and Management. University Park, Pennsylvania (2014)
2. Avila, L.A., Cangialosi, J.: National hurricane center. http://www.nhc.noaa.gov/data/tcr/AL092011_Irene.pdf (2011). Accessed July 2014
3. Bakshy, E., Rosenn, I., Marlow, C., Adamic, L.: The role of social networks in information diffusion. In: Proceedings of the 21st International Conference on World Wide Web WWW'12, pp. 519–528. ACM, New York (2012)
4. BBC News Europe: England riots: maps and timeline. http://www.bbc.co.uk/news/uk-14436499 (2011). Accessed July 2014
5. BBC News Europe: Timeline: how norway's terror attacks unfolded. http://www.bbc.co.uk/news/world-europe-14260297 (2012). Accessed July 2014
6. Bouchachia, A., Vanaret, C.: Incremental learning based on growing gaussian mixture models. In: 2011 10th International Conference on Machine Learning and Applications and Workshops (ICMLA), vol. 2, pp. 47–52 (2011)
7. Büscher, M., Liegl, M.: Connected communities in crises. In: Hellwagner, H., Pohl, D., Kaiser, R. (eds.) Social Media Analysis for Crisis Management, no. 1 in 2. IEEE Computer Society Special Technical Community on Social Networking E-Letter (2014)
8. Castillo, C., Mendoza, M., Poblete, B.: Predicting information credibility in time-sensitive social media. Internet Res. **23**(5), 560–588 (2013)
9. Choudhary, A., Hendrix, W., Lee, K., Palsetia, D., Liao, W.K.: Social media evolution of the Egyptian revolution. Commun. ACM **55**(5), 74–80 (2012)
10. Dashti, S., Palen, L., Heris, M.P., Anderson, K.M., Anderson, S., Anderson, S.: Supporting disaster reconnaissance with social media data: a design-oriented case study of the 2013 colorado floods. In: Proceedings of the 11th International Conference on Information Systems for Crisis Response and Management. University Park, Pennsylvania (2014)
11. Denef, S., Bayerl, P.S., Kaptein, N.: Social media and the police—tweeting practices of British police forces during the august 2011 riots. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'13), Paris, France (2013)
12. Dugdale, J., Van de Walle, B., Koeppinghoff, C.: Social media and SMS in the Haiti earthquake. In: Proceedings of the 21st International Conference Companion on World Wide Web WWW'12 Companion, pp. 713–714. ACM, New York (2012)
13. Fontugne, R., Cho, K., Won, Y., Fukuda, K.: Disasters seen through Flickr cameras. In: Proceedings of the Special Workshop on Internet and Disasters, SWID '11, pp. 5:1–5:10. ACM, New York (2011)
14. Hiltz, S.R., van de Walle, B., Turoff, M.: The domain of emergency management information. In: van de Walle, B., Truoff, M., Hiltz, S.R. (eds.) Information Systems for Emergency Management, vol. 16, pp. 3–19. Armonk, New York (2010)
15. Hughes, A.L., Palen, L.: Twitter adoption and use in mass convergence and emergency events. Int. Conf. Inform. Syst. Crisis Response Manag. (ISCRAM) **6**(3/4), 248 (2009)
16. Hughes, A., Denis, L.S., Palen, L., Anderson, K.: Online public communications by police and fire services during the 2012 Hurricane Sandy. In: Proceedings of the ACM 2014 Conference on Human Factors in Computing Systems (CHI), Toronto (2014)
17. Ireson, N.: Local community situational awareness during an emergency. In: 3rd IEEE International Conference on Digital Ecosystems and Technologies, 2009. DEST '09, pp. 49–54 (2009)
18. Klein, B., Laiseca, X., Casado-Mansilla, D., López-de-Ipiña, D., Nespral, A.P.: Detection and extracting of emergency knowledge from twitter streams. In: Bravo, J., López-de-Ipiña, D., Moya, F. (eds.) Ubiquitous Computing and Ambient Intelligence, Lecture Notes in Computer Science, pp. 462–469. Springer, Heidelberg (2012)

19. Kumar, S., Morstatter, F., Liu, H.: Monitoring social media for humanitarian assistance and disaster relief. In: Hellwagner, H., Pohl, D., Kaiser, R. (eds.) Social Media Analysis for Crisis Management, vol. 2. IEEE Computer Society Special Technical Community on Social Networking E-Letter (2014)
20. Liu, S., Palen, L., Sutton, J., Hughes, A., Vieweg, S.: In search of the bigger picture: the emergent role of on-line photo-sharing in times of disaster. In: Proceedings of the International Conference on Information Systems for Crisis Response and Management (ISCRAM) (2008)
21. Mendoza, M., Poblete, B., Castillo, C.: Twitter under crisis: can we trust what we RT?. In: Proceedings of the First Workshop on Social Media Analytics, SOMA'10, pp. 71–79. ACM, New York (2010)
22. Morstatter, F., Kumar, S., Liu, H., Maciejewski, R.: Understanding Twitter data with TweetXplorer. In: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD'13, pp. 1482–1485. ACM, New York (2013)
23. Netten, N., van Someren, M.: Identifying segments for routing emergency response Dialogues. In: 5th International Conference on Information Systems for Crisis Response and Management (ISCRAM) (2008)
24. Netten, N., van Someren, M.: Improving communication in crisis management by evaluating the relevance of messages. J. Conting. Crisis Manag. **19**(2), 75–85 (2011)
25. Palen, L.: Online social media in crisis events. EDUCAUSE Q. (EQ) **31**(3), 76–78 (2008). http://www.educause.edu/
26. Perng, S.Y., Büscher, M., Wood, L., Halvorsrud, R., Stiso, M., Ramirez, L., Al-Akka, A.: Peripheral response: microblogging during the 22/7/2011 Norway attacks. Int. J. Infor. Syst. Crisis Response Manag. (IJISCRAM) **5**(1), 41–57 (2013)
27. Pohl, D., Bouchachia, A., Hellwagner, H.: Automatic identification of crisis-related sub-events using clustering. In: 11th International Conference on Machine Learning and Applications (ICMLA), vol. 2, pp. 333–338 (2012)
28. Pohl, D., Bouchachia, A., Hellwagner, H.: Automatic sub-event detection in emergency management using social media. In: First Inter. Workshop on Social Web for Disaster Management (SWDM), In conjunction with WWW'12, Lyon, France (2012)
29. Pohl, D., Bouchachia, A., Hellwagner, H.: Supporting crisis management via sub-event detection in social networks. In: International Conference on Collaboration Technologies and Infrastructures, Toulouse, France (2012)
30. Pohl, D., Bouchachia, A., Hellwagner, H.: Online processing of social media data for emergency management. Int. Conf. Mach. Learn. Appl. (ICMLA) **2**, 408–413 (2013)
31. Pohl, D., Bouchachia, A., Hellwagner, H.: Social Media for Crisis Management: Clustering Approaches for Sub-Event Detection. Multimedia Tools and Applications, pp. 1–32 (2013)
32. Pohl, D., Bouchachia, A., Hellwagner, H.: Supporting crisis management via detection of subevents in social networks. Int. J. Inf. Syst. Crisis Response Manag. (IJISCRAM) **5**(3), 20–36 (2013)
33. Pohl, D., Bouchachia, A., Hellwagner, H.: Crisis-related sub-event detection based on clustering. In: Hellwagner, H., Pohl, D., Kaiser, R. (eds.) Social Media Analysis for Crisis Management, 1. IEEE Computer Society Special Technical Community on Social Networking E-Letter (2014)
34. Public Health Emergency: Public health and medical emergency support for a nation prepared: 2011 Mississippi river flooding. http://www.phe.gov/emergency/news/sitreps/Pages/2011mississippi-flooding.aspx (2011). Accessed July 2014
35. Public Health Emergency: Public health and medical emergency support for a nation prepared: Hurricane Irene 2011. http://www.phe.gov/emergency/news/sitreps/Pages/irene-2011.aspx (2011). Accessed July 2014
36. Purohit, H., Bhatt, S., Hampton, A., Shalin, V., Sheth, A., Flach, J.: With whom to coordinate, why and how in the ad-hoc social media communities during crisis response. In: Proceedings of the 11th International Conference on Information Systems for Crisis Response and Management. University Park, Pennsylvania (2014)

37. Raffelsberger, C., Hellwagner, H.: Combined mobile Ad-Hoc and delay/disruption-tolerant routing. In: Guo, S., Lloret, J., Manzoni, P., Ruehrup, S. (eds.) In: Proceedings of the 13th International Conference on ad-hoc, Mobile, and Wireless Networks (ADHOC-NOW), Lecture Notes in Computer Science (LNCS 8487), vol. 8487, pp. 1–14. Springer, Heidelberg (2014)
38. Ren, D., Zhang, X., Wang, Z., Li, J., Yuan, X.: WeiboEvents: a crowd sourcing Weibo visual analytic system. In: IEEE Pacific Visualization Symposium (PacificVis), pp. 330–334 (2014)
39. Reuter, C., Heger, O., Pipek, V.: Social media for supporting emergent groups in crisis management. In: Proceedings of the CSCW Workshop on Collaboration and Crisis Informatics, no. 2 in International Reports on Socio Informatics, pp. 84–92 (2012)
40. Reuter, C., Marx, A., Pipek, V.: Crisis management 2.0: towards a systematization of social software use in crisis situations. Int. J. Inf. Syst. Crisis Response Manag. (IJISCRAM) **4**(1), 1–16 (2012)
41. Sakaki, T., Okazaki, M., Matsuo, Y.: Earthquake shakes twitter users: real-time event detection by social sensors. In: Proceedings of the 19th International Conference on World Wide Web, WWW'10, pp. 851–860. ACM, New York (2010)
42. Starbird, K., Palen, L.: (How) will the revolution be retweeted?: information diffusion and the 2011 Egyptian uprising. In: Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work, CSCW'12, pp. 7–16. ACM, New York (2012)
43. Stollberg, B., de Groeve, T.: The use of social media within the global disaster alert and coordination system (GDACS). In: Proceedings of the 21st International Conference Companion on World Wide Web, WWW'12 Companion, pp. 703–706. ACM, New York (2012)
44. Sutton, J., Spiro, E., Fitzhugh, S., Johnson, B., Gibson, B., Butts, C.T.: Online message amplification in the boston bombing response. In: Proceedings of the 11th International Conference on Information Systems for Crisis Response and Management. University Park, Pennsylvania (2014)
45. Terpstra, T., de Vries, A., Stronkman, R., Paradies, G.L.: Towards a realtime twitter analysis during crises for operational crisis management. In: Proceedings of the 9th International Conference on Information Systems for Crisis Response and Management (ISCRAM), Vancouver, Canada (2012)
46. Tyshchuk, Y., Li, H., Ji, H., Wallace, W.A.: Evolution of communities on twitter and the role of their leaders during emergencies. In: Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM'13, pp. 727–733. ACM, New York (2013)
47. Vieweg, S., Hughes, A.L., Starbird, K., Palen, L.: Microblogging during two natural hazards events: what twitter may contribute to situational awareness. In: Proceedings of the 28th International Conference on Human Factors in Computing Systems, CHI'10, pp. 1079–1088. ACM, New York (2010)
48. Wei, J., Bu, B., Guo, X., Gollagher, M.: The process of crisis information dissemination: impacts of the strength of ties in social networks. In: Kybernetes, no. 2 in 43, pp. 178–191. Emerald Group Publishing Limited (2014)
49. Wikipedia Article: Effects of Hurricane Sandy in New York. http://en.wikipedia.org/wiki/Effects_of_Hurricane_Sandy_in_New_York. Accessed July 2014
50. Yates, D., Paquette, S.: Emergency knowledge management and social media technologies: a case study of the 2010 Haitian earthquake. Int. J. Inf. Manag. **31**(1), 6–13 (2011)
51. Zhou, Y., Yang, L., Van de Walle, B., Han, C.: Classification of microblogs for support emergency responses: case study Yushu earthquake in China. In: 2013 46th Hawaii International Conference on System Sciences (HICSS), pp. 1553–1562 (2013)

# Chapter 13
# Simulations of Financial Contagion in Interbank Networks: Some Methodological Issues

**Mario Eboli**

**Abstract** This contribution focuses on the methodology applied in papers that investigate the dynamics of contagion in financial networks using numerical simulations. In these papers, a propagation of losses and defaults in a financial system is modeled as a direct balance-sheet contagion (a.k.a. counterparty contagion), that is the direct transmission of losses from financially distressed debtors to their creditors. The researchers in this field perform their simulations with three different methods: (i) basic linear threshold algorithms, (ii) the graph-theoretic approach, where contagion is modeled as a propagation process in directed and weighted graphs, (iii) the lattice-theoretic approach, where contagion is modeled as a 'fictitious default algorithm', that computes the vector of payments that clears a net of financial obligations. Some of the results obtained by this stream of literature raised doubts about the assumptions used in such simulations. We discuss this issue and present some methodological recommendations that may improve the realism and the generality achievable in numerical investigations of financial contagion.

## 13.1 Introduction

Even before the financial turmoil that started in 2008, the concern for the risk of financial contagion prompted a growing number of economists to investigate the propagation of losses and defaults in networks of financially connected agents. Financial and banking networks arise from several contexts, such as the interbank liquidity market, where banks cross-hold short term liabilities in order to share liquidity risk; payment systems; over-the-counter markets for derivatives; the sharing of credit risks through syndicated loans, etc. In all these cases, there are sets of financial operators whose balance sheets are directly and indirectly connected to one another by financial obligations: an assets in the balance sheets of an agent is a liability in the balance sheet of another agent. In such financial networks contagion can occur through different and non alternative channels. First, and most important with respect to the focus of this chapter, default contagion can occur through the direct transmission of losses

M. Eboli (✉)
Dipartimento di Economia Aziendale, Universitá 'G. d'Annunzio', Pescara, Italy
e-mail: m.eboli@unich.it

D. Król et al. (eds.), *Propagation Phenomena in Real World Networks*,
Intelligent Systems Reference Library 85, DOI 10.1007/978-3-319-15916-4_13

from the liability side of the balance sheet of a defaulting debtor to the asset side of the balance sheets of its creditors. Starting from an initial set of defaults, the *primary* defaults, this transmission of losses can cause further, *secondary* defaults, if the losses received by an agent are larger that its absorbing capacity, i.e. its equity. This direct balance sheet process of default contagion, also known as *domino effect*, is the form of financial contagion that has been most investigated by economists, with both analytic and numerical methods. A second form of direct balance sheet contagion is the transmission of liquidity shortages. A lender that faces a liquidity deficit reduces the loans granted to its borrowers who, in turn, reduce their own exposures towards other agents, and so on. This illiquidity contagion is transmitted from the asset side of the balance sheet of a lender to the liability side of the balance sheets of its borrowers. Both these two forms of direct contagion unfold along the directed paths composed of the financial obligations that form the networks at hand; thus both have been modeled, and simulated, as network phenomena. Besides such direct, agent-to-agent transmissions of losses or illiquidity, contagion among agents that belong to a financial networks also occurs through the common depreciation of their assets, depreciations due to common or correlated exposures or due to the untimely sales of illiquid assets in scarcely liquid markets.[1]

In this contribution, we focus on papers that investigate direct balance-sheet default contagion by means of numerical simulations. This stream of literature can be divided in three parts. A number of economists, most of which work for central banks, have run simulations on national interbank networks in order to test their resiliency to possible shocks caused by the failure of one or more banks. These authors use datasets usually collected by the central banks (in their quality of monetary authorities), thus they work on real networks in a strict sense. A second group of authors, who study financial contagion with analytic methods, have run numerical simulations on randomly generated networks in order to complement and validate their arguments. Finally, another set of contributions characterise, mostly with mean-field approximation methods, the probability of the occurrence of *cascades* of defaults in financial networks, and use numerical simulations on randomly generated networks to validate the approximate results obtained with their approach. These second and third groups of contributions use simulations on stochastically generated networks, which are designed to capture some essential features of the observed interbank networks. Thus such networks are not real, in a strict sense, but realistic (plausible) representations of real networks. This literature is briefly reviewed in the next section. The rest of the chapter is organised as follows. In Sect. 13.3, we present a model of a financial network, based on balance sheet data, and the three procedures used to simulate default contagion in such a network. In Sect. 13.4, we voice our concerns about some methodological choices made in the existing numerical simulation of financial contagion and make some recommendations. Conclusions are drawn in Sect. 13.5.

[1] A sharp increase in the sales of an illiquid asset (e.g., a share or a long term bond) may push its price below the fundamental (true) value of the asset. This phenomenon is known as ‘liquidity pricing’. During a crisis, financial intermediaries can be forced to sell assets in response to liquidity shortages and/or excessive leverage (the so called ‘fire sales’), facing losses due to such liquidity pricing.

## 13.2 Analytic and Numerical Studies of Financial Contagion

Starting from the mid 90s, scholars of economic theory studied the risk of financial contagion, i.e. the risk that the financial distress of a bank can be transmitted to other banks through the network of financial obligations that connects them. Seminal early theoretical contributions, due to authors such as [4, 18, 31], analyse simple stylised interbank deposit networks to evaluate the effect that the shape of a network has on its resiliency to external shocks and to bank runs. [18, 31] analyzed the possibility of bank runs caused by changes of depositors' expectations about the solvency of a bank. Other papers, such as [4], present models in which financial crises arise as a consequence of downturns in the economic cycle.[2] In general, in this class of models, if the total available liquidity is sufficient to satisfy the liquidity need caused by an informational shock or a solvency shock, then interbank loans are an effective way to redistribute liquidity among banks and to share the liquidity risk. Conversely, in the opposite case of an aggregate shortage of liquidity, the existence of a net of financial obligations among banks creates the grounds for the diffusion of financial distress and for the occurrence of systemic crises.

This early analytic literature focused on stylised interbank networks usually composed of four banks and arranged in four shapes: complete networks, where each agent has obligations towards all other agents; partially connected networks; star shaped networks, also known as money centers, where there is a central node connected with all other (peripheral) nodes, which in turn are connected only with the center; and circle shaped networks. The formal analysis of systemic risk in more realistic financial networks proved to be difficult and, as a consequence, many authors turned to numerical simulations to investigate financial contagion in actual, real world interbank networks. Authors such as [13, 17, 20, 29, 34, 36],[3] who are mostly central bankers, have studied national interbank networks—constructed on the basis of banks' balance sheet data collected by national authorities—with the aim of evaluating their exposure to the risk of contagion. These authors stress test such interbank networks with counterfactual simulations, generally performed making one bank fail because of exogenous causes. All these studies indicate that, in such national banking systems, the risk of default contagion due to interbank exposures is very low.

Other authors—e.g. [30] and [32]—investigate financial contagion by running similar numerical simulations on randomly generated financial networks, rather than empirically observed networks. Theoretical contributions, such as [1], also present numerical simulations of financial contagion in stochastically generated networks, used by the authors to illustrate and support their analytic results.

Finally, another group of authors—namely [5, 6, 11, 21, 22, 25, 28]—run numerical simulations of financial contagion to validate the approximate results, based on

[2] Recessions can cause losses in the value of the assets held by banks, losses capable of rendering them insolvent. If depositors foresee the recession, they will protect themselves from possible bank defaults by withdrawing their deposits and, in so doing, they create the conditions for the occurrence of a widespread crisis.

[3] See the review by Upper [34] and the papers cited therein.

applications of mean-field theory, that they obtain in assessing the probability of default cascades in financial networks.

## 13.3 Direct Balance-Sheet Contagion in Financial Networks

In what follows, we focus on a simple version of a financial network, where the assets held by the financial intermediaries are grouped in two broad categories: external and intra-network assets; similarly, agents' liabilities are sorted in two classes, external debts and intra-network debts, which are assumed to have the same seniority in liquidation procedures.[4]

Let $\Omega$ be a set of $n$ financial operators indexed by $i = 1...n$, and let $d_{ij} \in \mathcal{R}^+$, be the amount of debt, if any, that agent $i$ owes agent $j$, for $i, j = 1...n$ and $i \neq j$. Each agent in $\Omega$ is characterized by its own balance sheet:

| Assets | Liabilities |
|---|---|
| $c_i = \sum_j c_{ij}$ | $d_i = \sum_j d_{ij}$ |
| $a_i$ | $h_i$ |
|  | $e_i$ |

where, on the assets side, $a_i \in \mathcal{R}^+$ is the value of the *external assets* owned by $i$—i.e., claims on agents that do not belong to $\Omega$, and $c_i \in \mathcal{R}^+$ is the sum of the claims that $i$ holds against other agents in $\Omega$, i.e., $c_i = \sum_j c_{ij}$. On the liability side of the balance sheet, the intra-network debt of the $i$-th agent, $d_i \in \mathcal{R}^+$, is the sum of the liabilities issued by $i$ and held by other agents in $\Omega$, i.e., $d_i = \sum_j d_{ij}$; $h_i \in \mathcal{R}^+$ is the amount of obligations that $i$ has towards external financiers (in the form of bonds, deposits, etc.); and $e_i \equiv a_i + c_i - d_i - h_i$ is the net worth (equity) of the $i$-th agent. Finally, let $A = \{a^k\}$, $k = 1...m$, be a set of external assets such that each $a^k$ in $A$ appears in the balance sheet of at least one operator in $\Omega$, and let $a_i^k \in \mathcal{R}^+$ be the amount of asset $k$ held by agent $i$, if any. Note that the network structure of this financial system is embedded in its adjacency matrix $[d_{ij}]_{n,n}$.

### *13.3.1 The Linear Threshold Algorithm*

In many real world networks of socially interconnected agents, the behaviour of an agent is influenced by the behaviour of her 'neighbours', i.e. the set of agents directly connected to her. The sociologist Mark Granovetter put forward a linear threshold model of diffusion of innovative behaviours that captures this mechanism. In his seminal work [23], Granovetter postulates that an agent is induced to adopt an

[4] This network model can be easily generalised by adding different liabilities with different seniority, as it is done in [1].

innovation when she observes, in her neighbourhood, a number of adoptions larger than a given threshold value. The linear threshold algorithms, based on this model, iteratedly compute the adoptions (i.e., the change of state of nodes in the network) induced by an initial set of early adopters. Starting with [20], several economists[5] have applied this type of algorithms in simulations of defaults contagion in national banking systems.

To present this approach, we need some more notation. Let $\mathbf{z}$ be a vector composed of $n$ variables $z_i$, $i = 1, ..., n$,where $z_i$ takes on value 0 if the $i$-th bank is solvent, while it takes on value 1 if the $i$-th bank is insolvent. Let $b \in (0, 1]$ be a parameter, exogenous and fixed, that measures the *loss-given-default* suffered by the debtors of a defaulting bank, i.e. the quota of the claims of the creditors which is not recovered through the liquidation of the bank. Finally, let $\lambda = [\lambda_i | i = 1, ..., n]$ be the vector composed of the losses $\lambda_i$ received by the banks in $\Omega$. A bank is brought to bankruptcy if it receives losses larger than its own net worth $e_i$.

The linear threshold algorithm applied to financial contagion, also known as *sequential default algorithm*, consists of the following procedure:

1. For a given initial set of defaulting banks $\underline{\Omega}$, i.e. for a given initial vector $\mathbf{z}$, compute $\lambda = b\mathbf{z}[d_{ji}]_{n,n}$.
2. If $\lambda_i < e_i$ for all banks in $\Omega \backslash \underline{\Omega}$, then stop. If $\lambda_i > e_i$ for one or more banks in $\Omega \backslash \underline{\Omega}$, then update $\underline{\Omega}$ and start again from step 1.

This procedure is the one adopted by almost all the above cited papers that stress test national interbank systems. This is surprising, considering that this approach suffers some limits that stem from the fact that it does not make full use of the information embedded in the above described representation of a financial network. These limits are discussed below in Sect. 13.4.2.

### 13.3.2 The Graph-Theoretic Approach

The natural candidate for a formal representation of the above described financial system is a directed and weighted graph $N = (\Omega, D)$, where the agents in $\Omega$ are represented as nodes and the obligations that connect them are represented as a set of weighted and directed links $D = \{d_{ij} | i \neq j \ \text{ and } \ i, j = 1, ..., n\}$.

In a network $N$, a propagation of losses is generated by an exogenous shock, as a drop in the value of some assets in $A$, defined as follows. Let $b^k \in [0, 1]$ be a parameter that measures the fraction of the value of the asset $a^k$ which is lost, and let $[b^k]$, $k = 1...m$, be the vector composed of such parameters. An exogenous shock is an assignment of value to this vector where at least one of its components assumes a strictly positive value. The loss received by a node $i$, which owns a positive amount of asset $a^k$, is $b^k a^k_i$ and the total value of the external shock is $\sigma = \sum_A b^k a^k$.

[5] See [34].

The propagation of losses through the network is governed by the rules of *limited liability*, *debt priority* and *pro rata reimbursement*. Limited liability and debt priority imply that when a node suffers a loss, this loss is first absorbed by the net worth of the node, i.e. by the equity held by the shareholders of the bank. Only the residual loss, if any, is passed over to other nodes in $\Omega$. For each node $i$ in $\Omega$, let

$$\beta_i(\lambda_i) = \min\left(\frac{\lambda_i}{e_i}, 1\right) \tag{13.1}$$

be an *absorption function*, where $\lambda_i$ is the total loss born by the $i$-th node, as defined below, and the variable $\beta_i \in [0, 1]$ measures the share of net worth lost by the shareholders, who suffer an amount of losses equal to $\beta_i\, e_i$.

If the losses suffered by $i$ are larger than its initial net worth, then this node is insolvent and sends the residual loss, $\lambda_i - e_i$, to its creditors, i.e., to its direct descendants in $\Omega$: $j \in \Omega$ such that $l_{ij} \in L$, also said *children* nodes of $i$. For each node $i$ in $\Omega$, let

$$b_i(\lambda_i) = \max\left(0, \frac{\lambda_i - e_i}{d_i + h_i}\right) \tag{13.2}$$

be a *loss-given-default function*. The variable $b_i \in [0, 1]$ assumes a value of zero if the $i$-th operator is solvent, while it assumes a strictly positive value if the operator defaults. In the latter case, the assets of the insolvent node are liquidated and its creditors get a *pro rata* refund.[6] Thus $b_i$ measures the fraction of the $i$-th agent's debt that can not be recovered through liquidation. When the $i$-th agent becomes insolvent, a node $j$ which is a creditor of node $i$ receives from the latter a loss equal to $b_i c_{ij}$. The total loss born by an agent in $\Omega$, if any, is the sum of the loss of value of its external assets plus the losses received from its defaulting parent nodes: $\lambda_i = \sum_k b_k a_i^k + \sum_j b_j c_{ji}$.

With this setting, the value of a propagation of losses and defaults in a financial network $N$ is computed through the iterated application—node by node, along the directed paths and cycles of $N$—of the absorption and loss-given-default functions defined above. For a given exogenous shock, this procedure yields a sequence of passing of financial losses starting from the initial set of defaults. When the initial exogenous shock has been completely absorbed by the portfolios of shareholders and external financiers, no further default can occur, the computation stops and the algorithm delivers the pair of $n$ dimensional vectors $\{[\beta_i], [b_i]\}$ which identify the propagation at hand. In this way the procedure characterises the losses born by each node and the final set of defaulting agents, for any given value of the shock vector $[b_k]$.

[6] Some authors assume that this is done without incurring bankruptcy costs, while other consider the presence of fixed or porportional liquidation costs and, finally, several authors simplify their analysis assuming that for each failing agent $b_i = 1$, i.e. assume that creditors get no refund at all.

Let us add a superscript $t = 1, 2, 3, \ldots$ to the variables involved in the computation—namely $\lambda_i^t$, $b_i^t$, $\beta_i^t$—to indicate the value taken on by these variables at each iteration of the algorithm. Recall that $\lambda_i = \sum_k b_k a_i^k + \sum_j b_j c_{ji}$ and let

$$[\lambda_i]_{1\times n} = [b_k]_{1\times m} \left[a_i^k\right]_{m\times n} + \left[b_j\right]_{1\times n} \left[d_{ji}\right]_{n\times n} \tag{13.3}$$

be the vector of the losses born by the agents in $\Omega$. Further, let $\underline{\Omega} = \{i \in \Omega | b_i > 0\}$ be the set of insolvent agents. The algorithm is the following:

1. For the given value assignment of the vector $[b^k]$, compute $[\lambda_i^t] = [b_k] \left[a_i^k\right] + \left[b_j^{t-1}\right] \left[d_{ji}\right]$, starting with $t = 1$ and setting $[b_j^0] = [0]$.
2. Compute $[\beta_i^t] = [\beta_i(\lambda_i^t)]$ and $[b_i^{t-1}] = [b_i(\lambda_i^t)]$ according to 13.1 and 13.2.
3. If $\sum_\Omega \beta_i^t e_i + \sum_\Omega b_i^t h_i = \sum_A b_k a_k$, then stop. If $\sum_\Omega \beta_i^t e_i + \sum_\Omega b_i^t h_i < \sum_A b_k a_k$ , then start again from step 1.[7]

The values of the vectors $[\lambda_i^t]$, $[\beta_i^t]$, $[b_i^t]$ are strictly increasing in $t$ as long as there are nodes in $\Omega$ with strictly positive divergence (incoming losses larger than outgoing losses), i.e., as long as there exists at least one $i \in \Omega$ s.t. $\lambda_i^t > \beta_i^{t-1} e_i + b_i^{t-1} d_i$, which, in turn, implies $\sum_\Omega \beta_i^t e_i + \sum_\Omega b_i^t h_i < \sum_A b_k a_k$ . Conversely, the repeated iteration of the algorithm yields stationary values of the vectors at hand once $\sum_\Omega \beta_i^t e_i + \sum_\Omega b_i^t h_i = \sum_A b_k a_k$. This condition is eventually achieved, then the divergence of all nodes in $\Omega$ is null and neither the losses arriving at a node nor the losses departing from a node can grow anymore.

Each iteration of this algorithm computes the passing of losses from a set of nodes in $N$ to their children nodes, i.e., to their direct descendants. In absence of cycles, the length of the longest possible path in $N$ is equal to $n$ and so is the largest possible number of iterations in the algorithm. Conversely, in presence of cycles of defaulting nodes, the algorithm converges asymptotically to the solution values by computing progressively smaller augmentations of the losses passed along the cycle. Since the values at hand are sums of money, this problem can be easily overcome by discretising the variables involved. In this case the algorithm stops in a finite number of iterations.

This procedure for the computation of a contagion process has been applied in different contexts. [30][8] uses this approach, and the above algorithm, in simulations run on randomly generated financial networks which are parameterised with respect to four features of a banking sector: the capacity of banks to absorb shocks (i.e., their capital), the size of interbank exposures, the degree of connectivity, and the degree of concentration. They run simulations to evaluate the effects that these characteristics of a banking sector have on its resiliency to external shocks. [27] uses a very similar

[7] This terminal condition can also be expressed in terms of defaulting nodes, as it is done in the linear threshold algorithm depicted above, because the condition $\sum_\Omega \beta_i^t e_i + \sum_\Omega b_i^t h_i = \sum_A b_k a_k$ implies that no more nodes default, and vice versa.

[8] These authors build their model on an early version of [14], presented at the bank of England in May 2004. See [30],p. 2038.

algorithm to simulate the propagations of systemic liquidity shortages in differently shaped interbank networks. The above procedure is also used in the numerical simulations performed by most of the authors of the default cascade models discussed below in section.

[15] develops the graph theoretic model described so far, transforming it into a flow network. In this paper, the above financial system is represented as a *multi-source flow network*, i.e., a directed and connected graph, with some sources and some sinks, whose links are endowed with non-negative capacities.[9] This is done by adding to the above graph $N = (\Omega, L)$ a set $A = \{a^k\}$ of *m source nodes*—i.e., nodes with no incoming links and with outgoing links ending in the nodes in $\Omega$—that represent the above defined external assets, and two *sinks*—i.e., terminal nodes with no outgoing links and with incoming links starting from the nodes in $\Omega$—that represent the portfolios of the final claimants of the financial system, divided in two groups: shareholders and debtholders. This transformation enabled us to model a contagion process as a flow of losses that crosses a flow network, starting from the source nodes in $A$ and ending in the sink nodes. In [15] we use the properties of such flows to asses and compare the exposure to contagion of complete, star-shaped, circular and regular incomplete networks; and to establish some properties of contagion processes in generic networks. This flow-network approach is also applied by [9], who study the dynamics of liquidity flows in interbank networks in order to identify the network shape that provides coverage from liquidity risk with the smallest possible exposure to contagion risk.

### 13.3.3 The Lattice-Theoretic Approach

Payment systems form networks of obligations that are a potential channel of default contagion among the participants to the system. In their seminal work, Eisenberg and Noe [16] introduce a lattice-theoretic[10] model of a payment system where (i) the participants are connected among themselves by a net of nominal liabilities (promised payments), and (ii) each participant receives a positive operating cash flow. In compliance with the rules of limited liability and pro-rata reimbursements of creditors, an agent that receives a cash inflow (which is the sum of her own operating cash flows plus the payments received by other agents in the system) smaller that her payment obligations, defaults on her creditors who, in turn, receive a pro-rata quota of the cash inflow of the defaulting agent. Eisenberg and Noe show that for a given payment system, composed of a vector of operating cash flows and a matrix of nominal liabilities, there always exists a vector of payments that clears the net of

[9] See [3], for the definition of a multisource flow network and for its properties.

[10] A lattice is a system $\langle X, \leq \rangle$ composed of a non-empty set $X$ and a binary relation $\leq$, where the latter induces a partial ordering on the elements of $X$ and, for any two elements $x, y \in X$, there exists a least upper bound (a.k.a. *join* or *supremum*) $x \vee y$ and a greatest lower bound (a.k.a. *meet* or *infimum*) $x \wedge y$.

obligations. They also show that, under a mildly restrictive condition, such a vector is unique. In order to identify a clearing payments vector, Eisenberg and Noe present an algorithm, the *fictitious default algorithm*, that iteratively computes the propagation of defaults (if any) for a given value assignment to the exogenous earnings of the agents. This model of systemic risk based on clearing payment flows, and its fictitious default algorithm, are directly applicable to the above balance-sheet representation of a financial system.[11]

Keeping the above notation, let $\overline{x}_i = \sum_j d_{ij} + h_i$ be the nominal obligations of a node $i$ in $\Omega$, let $\Pi = [\pi_{ij}]_{n,n}$ be the *relative liability matrix* of $N$, where $\pi_{ij} = d_{ij}/x_i$, and let $x_i$ be the sum of the payments made by agent $i$. Thus the payment of agent $i$ to agent $j$ is equal to $x_i\ \pi_{ij}$ and the total payments that $i$ receives from other agents in $\Omega$ are $\sum_j \pi_{ji}\, d_{ji}$. If agent $i$ is solvent, she pays in full her obligations and her creditor $j$ receives $\overline{x}_i\ \pi_{ij}$ . If, conversely, agent $i$ is insolvent, i.e. $\overline{x}_i < a_i + \sum_j \pi_{ji}\, d_{ji}$, she pays out $\pi_{ij}\left(a_i + \sum_j \pi_{ji}\, d_{ji}\right)$ to her creditor $j$ in $\Omega$. Hence the total payment of agent $i$ to the other agents in the network is

$$x_i = \min\left(\overline{x}_i, a_i + \sum_j \pi_{ji} d_{ji}\right)$$

This holds for all $i \in \Omega$ and can be written as

$$\mathbf{x} = \overline{\mathbf{x}} \wedge \left(\mathbf{a} + \Pi^T \mathbf{x}\right) \tag{13.4}$$

where: $\Pi^T$ is the transpose of $\Pi$; $\mathbf{x} = [x_i]$, $\overline{\mathbf{x}} = [\overline{x}_i]$ and $\mathbf{a} = [a_i]$ are, respectively, the vector of payments, the vector of nominal obligations and a value assignment to the vector of external assets. The vector of payments $\mathbf{x}$ that clears the system is the solution of the mapping $H(\cdot|\mathbf{a}, \overline{\mathbf{x}}, \Pi) : [0, \overline{\mathbf{x}}] \rightarrow [0, \overline{\mathbf{x}}]$ defined as

$$H(\mathbf{x}|\mathbf{a}, \overline{\mathbf{x}}, \Pi) \equiv \overline{\mathbf{x}} \wedge \left(\mathbf{a} + \Pi^T \mathbf{x}\right)$$

Using Tarski's fixed point theorem, Eisenberg and Noe demonstrate that this increasing and bounded map has a solution and that such a solution is unique under a mildly restrictive assumption. To characterise this clearing payment vector, the authors use what they called the *fictitious default algorithm*, that consists of the following procedure. For a given vector of external assets values—i.e., for a given exogenous shock that induces an initial set of defaults:

1. Compute $\mathbf{x} = \overline{\mathbf{x}} \wedge \left(\mathbf{a} + \Pi^T \mathbf{x}\right)$ assuming that all non-defaulting banks honour their obligations in full.
2. Stop if no further bank defaults. If there are new defaults, start again from step 1.

[11] The following description of the lattice-theoretic approach is inspired to [10], which is one of the papers that applies the Eisenberg and Noe model.

In a system composed of $n$ banks, this procedure stops in at most $n$ iterations and delivers both the clearing payment vector and the sequence of defaults induced by the exogenous shock.

This algorithm has been applied in numerical simulations of contagion in national banking systems only by [16].[12] Conversely, several theoretical papers on financial contagion—such as [1, 2, 10, 32]—are built on the basis of the Eisenberg Noe model. [1], inter alia, use the fictitious default algorithm in simulations of contagion in randomly generated networks.

## 13.4 Some Methodological Issues

As Upper points out [34], the results obtained by the above cited simulations of contagion in national interbank networks are strongly affected by the "potential bias caused by the very strong assumptions underlying the simulations." Upper indicates a list of nine assumptions that need to be carefully evaluated, in particular with respect to their degree of realism, when applied to simulations of financial contagion. The most relevant of these suggestions, in our opinion, are the ones that concern two assumptions, namely the assumption that non-interbank liabilities are senior with respect to interbank liabilities and the assumption that external (non liquid) assets can be sold at their book value.

The rules that allocate the losses among the creditors of an insolvent agent play a crucial role in a default contagion process. Assuming that the claims held by agents who do not belong to the network, i.e. the external financiers, are senior, in liquidation procedures, with respect to interbank claims, amplifies both the possibility and the scope of default contagion.[13] This is due to the fact that this restriction increases the share of losses that circulate within the financial intermediaries and reduces the amount of losses absorbed by the external financiers. Moreover, there is no clear evidence in support of the realism of this assumption.[14]

The assumption that long-term assets can be liquidated at book value rules out an important channel of contagion: the price effects of 'fire sales' of illiquid assets sold in scarcely liquid markets. This simplifying restriction leads to underestimate the possibility and the magnitude of contagion during a crisis, since the liquidations of assets forced by bankruptcies and by the de-leveraging undertaken by banks are typical phenomena of periods of financial turmoil.

We share the concerns raised by Upper and, in what follows, we add to them our reflections on the implications of two methodological choices made in the simulations of financial contagion: the choice of the algorithm used to run the simulations and the assumptions made about the size and the distribution of the exogenous shocks

[12] See [34].

[13] For instance [30], assume that customer deposits are senior to interbank liabilities, while there is no evidence in support of such a restriction.

[14] See [34].

used in the simulations. Moreover, we discuss the implications of the results of the contributions that run simulations to validate the approximate analytic methods used to evaluate the probability of the occurrence of default cascades.

### *13.4.1 The Choice of the Algorithm*

As pointed out by Upper [34], most of the authors that performed numerical simulations of contagion in national banking systems choose to use the sequential default algorithm. This choice is questionable because this algorithm has two main drawbacks, while we see no specific gains from its usage. First, this approach has no room for an accounting of the changes of value of the claims that banks hold against non financial agents, i.e. the exposures of the banking system towards the rest of the economy: the external assets $\mathbf{a}$. Since external assets are not explicitly present in the model, exogenous shocks can be modelled only as the default of one or more banks[15] with a given and fixed loss-given-default rate. Moreover, this feature also prevents the analyser from expanding the model to encompass other forms of contagion, such as the presence of common exposures (e.g., a sovereign debt, such as greek bonds, held by many European banks), the occurrence of 'fire sales' and of their feedback effects on the value of the assets which are 'marked-to-market'[16] in the balance sheets of banks. Second, the sequential default algorithm uses an exogenous and fixed loss-given-default for each defaulting bank. This simplification erases, from the performed simulations, the second order effects of the contagion process: Whenever a default propagation entails a directed cycle of defaulting agents, there is a feedback effect on the losses that are transmitted along the cycle, increasing the loss-given-default of the involved banks.[17] In other words, the intercyclicity of obligations, that generally exists in financial networks, magnifies the flows of losses passed along cycles of defaulting agents. This mechanism also introduces a simultaneity of the solution values of the loss-given-default of the banks that belong to cycles, a simultaneity that is not modeled by the sequential default algorithm. It may be argued that the magnitude of such second order effects can be, in reality, negligible.[18] It can also be argued that the time required for the liquidation of a bankrupt bank is long and that, therefore, the assumption of a loss-given-default equal or close to unity is a good approximation—in the time span of a contagion—of the impact of a defaulting bank on its creditors. Nonetheless, we see no reason why the loss-given-default rates of the banks should not be treated as endogenous to the model, as they indeed are, given that there are no technical difficulties in so doing. In fact, the graph-theoretic and the lattice-theoretic approaches described above do not face any

[15] Most of the above cited authors perform numerical simulations of the effects of the default of a single bank. Again, see [34].

[16] As opposed to historical value.

[17] More precisely, feedbacks of losses, in a default contagion process, take place in strongly connected components of a graph $N = (\Omega, D)$ defined below, if all member of the component default.

[18] See below Sect. 13.4.3.

of these limitations. Therefore we recommend the use of one of them in numerical simulations of financial contagion.

### *13.4.2 The Assumptions Made About the Exogenous Shocks*

As mentioned above, most of the authors that stress test national banking systems use, as external shocks, the failure of single banks. According to [34], only [17] and [19] model the possibility of multiple failures due to common shocks, with methods that aim to estimate the joint probability distribution of the value of the exposures of the banks (the external assets in $A$). [34] notes this scarce attention to common shocks and says that "the focus on idiosyncratic failures reveals a worrisome lack of thinking about the scenarios underlying the simulations" [34]. We share this view and believe that the fact that financial networks can be "robust-yet-fragile" renders essential to consider, in simulations of financial contagion, the possibility of scenarios that imply multiple failures accompanied by a general weakening of the system.

The "robust-yet-fragile" nature of financial systems was first conjectured by [24], who argued that densely connected financial networks may "exhibit a knife-edge or tipping point property", in the sense that "within a certain range, connections serve as shock-absorbers [and] connectivity engenders robustness." While, if the system is perturbed by sufficiently large shocks, high connectivity becomes the channel that enables a pervasive diffusion of defaults. Using analytic methods, [1] and [14] established that complete networks display a phase transition behaviour in the response to shocks of different magnitude: up to a certain threshold, these networks are completely resilient to shocks, in the sense that no secondary defaults occur; beyond such threshold, the entire network defaults. The rationale for this result lies in the fact that, in complete networks, the losses (and/or the illiquidity) are evenly transmitted by the financially distressed nodes to all other nodes, making the most of the absorbing capacity of the system as a whole. For the same reason, shocks which are larger than the absorbing capacity of the network cause its complete collapse. [15] shows that star-shaped networks have the same property, because if the central node defaults, the losses are equally born by all peripheral nodes.

In the light of these analytic results, it is not surprising that all the above cited works, that stress tested national systems with respect the initial default of a single bank, concluded that national systems are exposed to little risk of direct contagion. It is plausible that the "robust-yet-fragile" property belongs also to the empirically observed interbank networks, in as much as they appear as a combination of complete and star-shaped networks. National interbank networks often display a two-tiered, core-periphery disassortative structure, with heavy tails in the degree distributions.[19]

[19] Several studies agree on the observation of actual interbank lending networks formed by banks that consist of a core of highly connected banks, while the remaining peripheral banks are connect only to the core banks. [33] and [7] note this feature for the US commercial banks network. [8, 12, 26], respectively, find similar structures in banking networks of the UK, Canada, Japan, Germany and Austria.

In other words, they tend to appear as multi-star networks, where the money center nodes tend to be large and highly connected among themselves, forming the core to which the peripheral nodes are preferentially connected.

For these reasons, we think that the modeling of shock scenarios (possibly based on estimates of the future value of the external assets) that may embrace entire sections of a financial system is a crucial requisite to achieve plausible simulations of financial contagion. Moreover, it is advisable that numerical exercises take also into account the above described effects of fire sales in order not to underestimate the actual threat posed by 'low probability—high impact' scenarios, that may be not so unlikely during a financial crisis.[20]

### 13.4.3 Existence of the Simulated Functions and the Scope for Approximations

The existence of the propagation processes described above in Sect. 13.3 is guaranteed under general conditions. As mentioned above, [16] demonstrate the existence of a clearing payment vector applying Tarski's fixed point theorem to the function that maps the vector of intra-network payments onto itself. In a flow network setting, [14] proves the existence of the above defined propagation flow, for any realization of the external shock, applying the known *minimum cut–maximum flow* theorem by Ford and Fulkerson. The fact that a propagation function, as well as a clearing payment vector, always exists in a financial network stems from the budget identities which, in turn, ensure that the Kirchoff's law (the inflow of a node must equal its outflow) is always respected in a financial network $N = (\Omega, D)$.

While the existence of the functions used to model the propagation process in a deterministic fashion is not an issue, the same cannot be said for models that aim to evaluate the *probability* of default contagion and of the occurrence of *cascades* of defaults, i.e. large systemic crisis triggered by an initial small shock. Several contributions present analytic solutions to the problem of expressing these probabilities as a function of some parameters of the financial networks, parameters such as degree of connectivity (density) and of concentration, assortativity, capitalization and leverage of the banks, etc. These papers derive *probability generating functions,* that characterise the probability of default of nodes or of clusters of nodes in a financial network, with different but strictly related methods. [21, 22, 25] apply and extend the known cascade model by [35], which is based on mean-field approximations. [6, 28] use variational approximation methods, also based on mean-field theory. [5, 11, 25] apply the asymptotic properties of large, homogeneously sparse and locally 'tree-like' networks.

The need to resort to these approximation procedures can be explained as follows. In a financial network, for example, the probability of default of a node is

[20] See [32] for a model where fire sales and 'marking-to-market' of assets amplify the effects of contagion.

determined by the probability of default of its parent nodes,[21] i.e. its debtors, which, in turn, depends on the probability of default of their parent nodes, and so forth. In order to derive a closed-form, analytic function that captures the net of probabilistic dependences and independencies in a network, it is necessary that the probability of default of a node is suitable to be factorised in terms of the probabilities of default of its parent nodes: $p(x_i) = \Pi_i p(x_i|x_j, j \in V_i)$, where $x_i, x_j \in N$ and $V$ is the set of parent nodes (or of neighbours) of $x_i$.[22] In turn, such a factorisation is possible only if the probabilities of default of the parent nodes are independent on one another. This requirement becomes slightly less restrictive if the information embedded in the structure of $N$, i.e. in its adjacency matrix, is used to exploit the conditional independencies present in it.[23,24]

In finite graphs, the conditional independence condition is satisfied only in directed acyclic graphs, while the independence condition is satisfied only in trees, which are minimally connected directed acyclic graphs, where each node has just one parent node. In infinite graphs, the independence condition is asymptotically satisfied in large, homogeneously sparse and locally 'tree-like' graphs.[25] The probability generating functions derived in the cascade models at hand have an exact solution only when applied to these types of graphs. When these analytic measurements of the probability of contagion are applied to networks that entail cycles, then the results that they produce can only be taken as approximations that converge asymptotically to the true properties of such networks.

With the notable exception of [6], all the above cited authors explicitly discuss the approximate nature of the results obtained by applying their probability generating functions to finite networks which are not acyclic, and perform numerical simulations of contagion to test the reliability of their results. These simulations show that the analytic approximations work surprisingly well, in some cases *too* well.[26] It is reassuring, but not surprising, that such approximations yield reliable results for large

[21] Or of its neighbours, if the network is undirected.

[22] More precisely, the factorisation of the probability of the states of a node enables the analytic treatment of probabilities in networks, as it is done with Bayesian networks. To derive closed form solutions for such probabilities, it is necessary to impose further restrictions. In most of the above cited papers, the probability of default of a node depends solely on the number of defaulted debtors. In these cases the probability of default of the parent nodes can modeled as a binomial and, with this restriction, the probability of cascades is characterised in closed form. See, inter alia [6].

[23] For example, if the states of node $x$ and node $y$ are not independent of one another because they share a common ancestor, node $z$, then $x$ and $y$ are conditionally independent on one another given the state of node $z$. If, conversely, two or more of the parent nodes in $V_i$ belong to a directed cycle (or, more generally, to a strongly connected component of $N$), then the required independence condition cannot be satisfied, not even conditionally to the state of common ancestors.

[24] [25] exploit this property in their double (illiquidity and insolvency) cascade model.

[25] See [11, 25], for rigorous and detailed discussions of the role played by the independence conditions in the present context.

[26] Even the authors themselves appear surprised by their own results: "...numerical studies are in reassuring, some might say surprising, agreement with the results obtained from the analytic approximations..." [28]; "Extensive cross testing with Monte Carlo estimates shows that this approximate analysis performs surprisingly well." [22].

and sparse networks.[27] This is the case for a number of the tests performed by the above cited papers (e.g., [25] set $n = 20{,}000$ in some of their tests). What is amazing is that these analytic approximations seem to generate acceptable results even in the cases of small and dense networks. For instance [25], run a test on a dataset composed of 90 European banks and one on randomly generated networks with $n = 25$ and average degree equal to five—which are very small networks plenty of cycles. In both cases the authors obtain a surprising accordance between the results of the Monte Carlo simulations and the analytic approximations.

The point that we raise here is that this set of results can be seen as a test on the actual relevance of second order, cycle effects in contagion processes. In the light of the above discussion, we believe that these results should be tested under more general conditions, with two possible outcomes. Further testing may show that there is a bias in these exercises (we would expect to find an underestimation of local contagion phenomena, due to the negligence of the reinforcing effects of cycles) and hopefully help to clarify the scope and the limits of such approximations, identifying the types of networks more suitable for these exercises. If, conversely, further testing confirms the high reliability of these approximation procedures even for small and dense networks, then we can feel authorised to neglect the existence of cycles in financial networks without causing excessive harm to the analysis of financial contagion. This outcome would open the doors to possible applications, in this field, of known models of inference in directed acyclic graphs, such as Bayesian networks and other graph-theoretic representations of Markov fields.

## 13.5 Concluding Remarks

Numerical simulations proved to be a rather useful tool in investigating the properties of systems which are too complex to be studied with analytic methods. But, while an analytic result holds under conditions which are clearly stated, the same cannot be said for numerical exercises. As is known, analysers who run numerical simulations also run the risk of obtaining “simulation based” results, i.e. results that lack generality because they strongly depend on the assumptions and restrictions applied in the settings of the simulations. Numerical simulations of financial contagion processes do not escape this limit. Concerns have already been raised about the assumptions adopted in numerical simulations run on interbank networks. We join these methodological perplexities and add our recommendations about the algorithm used in these numerical exercises and about the assumptions made with respect to the shock scenarios. Finally, we notice the surprising performance of some analytic, closed form measures of the probability of default cascades, derived in the recent years by several authors on the basis of mean-field approximations. These analytic methods appear to deliver extremely reliable approximations—according to the authors, who tested

[27] As argued by [35], numerical results in random graph models approximate analytical solutions as $n$ gets close to 10,000.

their own results with vis-à-vis comparisons with numerical simulations run on the same networks and with the same shocks. This interesting phenomenon is a call for further numerical investigations aimed to test the reliability of these methods when they are applied to financial contagion processes. A confirmation of the apparent high reliability of these methods would open new direction of research, in as much as it would authorise the use of other methodologies, based on the graph-theoretic representations of stochastic domains (e.g. Bayesian networks), for the assessment of default and cascade probabilities in financial networks.

## References

1. Acemoglu, D., Ozdaglar, A.E.: Opinion dynamics and learning in social networks. Levine's working paper archive, David K. Levine (2010)
2. Afonso, G., Shin, H.S.: Systemic risk and liquidity in payment systems. Technical report 352, Federal Reserve Bank of New York, New York (2009)
3. Ahuja, R.K., Magnanti, T.L., Orlin, J.B.: Network Flows: Theory, Algorithms, and Applications. Prentice Hall, London (1993)
4. Allen, F., Gale, D.: Financial contagion. J. Polit. Econ. **108**(1), 1–33 (2000)
5. Amini, H., Cont, R., Minca, A.: Stress testing the resilience of financial networks. Int. J. Theor. Appl. Finance 15(1) (2012)
6. Battiston, S., Delli Gatti, D., Gallegati, M., Greenwald, B., Stiglitz, J.E.: Liaisons dangereuses: Increasing connectivity, risk sharing, and systemic risk. J. Econ. Dyn. Control **36**(8), 1121–1141 (2012)
7. Bech, M.L., Atalay, E.: The topology of the federal funds market. Technical report 354, Federal Reserve Bank of New York, New York (2008)
8. Boss, M., Elsinger, H., Summer, M., Thurner, S.: Network topology of the interbank market. Quant. Finance **4**(6), 677–684 (2004)
9. Castiglionesi, F., Eboli, M.: Liquidity Flows in Interbank Networks. Mimeo (2011)
10. Cifuentes, R., Ferrucci, G., Shin, H.S.: Liquidity risk and contagion. J. Eu. Econ. Assoc. **3**(2–3), 556–566 (2005)
11. Cont, R., Moussa, A., Santos, E.B.: Network structure and systemic risk in banking systems. In: Fouque, J.P., Langsam, J.A. (eds.) Handbook on Systemic Risk, pp. 327–368. Cambridge University Press, Cambridge (2013) (Cambridge Books Online)
12. Craig, B., von Peter, G.: Interbank tiering and money center banks. J. Financ. Intermed. **23**(3), 322–347 (2014)
13. Degryse, H., Nguyen, G.: Interbank exposures: an empirical examination of contagion risk in the belgian banking system. Int. J. Central Banking **3**(2), 123–171 (2007)
14. Eboli, M.: Systemic Risk in Financial Networks: A Graph Theoretic Approach. Universita di Chieti Pescara, Italy (2004)
15. Eboli, M.: A flow network analysis of direct balance-sheet contagion in financial networks. Technical report 1862, Kiel Institute for the World Economy, Germany (2013)
16. Eisenberg, L., Noe, T.H.: Systemic risk in financial systems. Manage. Sci. **47**(2), 236–249 (2001)
17. Elsinger, H., Lehar, A., Summer, M.: Risk assessment for banking systems. Manage. Sci. **52**(9), 1301–1314 (2006)
18. Freixas, X., Parigi, B.M., Rochet, J.C.: Systemic risk, interbank relations, and liquidity provision by the central bank. J. Money Credit Banking **32**(3), 611–638 (2000)
19. Frisell, L., Holmfeldt, M., Larsson, O., Omberg, M., Persson, M.: State-dependent contagion risk: using micro data from Swedish banks. Technical report, Sveriges Riksbank, Sweden (2007)

20. Furfine, C.: Interbank exposures: quantifying the risk of contagion. J. Money Credit Banking **35**(1), 111–128 (2003)
21. Gai, P., Kapadia, S.: Contagion in financial networks. Proc. R. Soc. A Math. Phys. Eng. Sci. **466**(2120), 2401–2423 (2010)
22. Gleeson, J.P., Hurd, T., Melnik, S., Hackett, A.: Systemic risk in banking networks without monte carlo simulation. In: Advances in Network Analysis and its Applications, vol. 18, pp. 27–56. Springer, boston (2013)
23. Granovetter, M.: Threshold models of collective behavior. Am. J. Sociol. **83**(6), 1420–1443 (1978)
24. Haldane, A.G.: Rethinking the financial network. Speech delivered at the Financial Student Association, Amsterdam
25. Hurd, T.R., Gleeson, J.P.: On watts cascade model with random link weights. J. Complex Netw. **1**(1), 25–43 (2013)
26. Inaoka, H., Ninomiya, T., Taniguchi, K., Shimizu, T., Takayasu, H.: Fractal network derived from banking transaction-an analysis of network structures formed by financial institutions. Bank of Japan working papers 4 (2004)
27. Lee, S.H.: Systemic liquidity shortages and interbank network structures. J. Financ. Stab. **9**(1), 1–12 (2013)
28. May, R.M., Arinaminpathy, N.: Systemic risk: the dynamics of model banking systems. J. R. Soc. Interface **7**(46), 823–838 (2010)
29. Mistrulli, P.E.: Assessing financial contagion in the interbank market: maximum entropy versus observed interbank lending patterns. J. Banking Finance **35**(5), 1114–1127 (2011)
30. Nier, E., Yang, J., Yorulmazer, T., Alentorn, A.: Network models and financial stability. J. Econ. Dyn. Control **31**(6), 2033–2060 (2007)
31. Rochet, J.C., Tirole, J.: Interbank lending and systemic risk. J. Money Credit Banking **28**(4), 733–762 (1996)
32. Shin, H.S.: Risk and liquidity in a system context. J. Financ. Intermed. **17**(3), 315–329 (2008)
33. Soramäki, K., Bech, M.L., Arnold, J., Glass, R.J., Beyeler, W.E.: The topology of interbank payment flows. Physica A **379**(1), 317–333 (2007)
34. Upper, C.: Simulation methods to assess the danger of contagion in interbank markets. J. Financ. Stab. **7**(3), 111–125 (2011)
35. Watts, D.J.: A simple model of global cascades on random networks. Proc. Nat. Acad. Sci. **99**(9), 5766–5771 (2002)
36. Wells, S.: U.K. interbank exposure: systemic risk implications. Financ. Stab. Rev. **13**, 175–182 (2002)

# Chapter 14
# Maximizing Social Influence in Real-World Networks—The State of the Art and Current Challenges

**Radosław Michalski and Przemysław Kazienko**

**Abstract** The following chapter aims to present the current research in the area of modelling and maximizing social influence in networks. Apart from describing the most popular models for this process, it focuses on presenting the advances in maximizing the spread of influence in social networks. Since most of the research was suited for static networks case, nowadays it is necessary to move it toward the networks that are everywhere around us—the dynamic ones. As is widely agreed in the scientific community, static networks are unacceptable simplification of the real world processes, so current research is moving toward the temporal networks. It is especially important when modelling propagation phenomena, such as the spread of influence, epidemics or diffusion of innovations. In this chapter it is presented how the research on maximizing the spread of influence is starting to explore real-world cases and how the early attempts of solving this problem for temporal networks look like. Moreover, it is shown how to benefit from the temporal properties of the social network in order to achieve better results for spread of influence compared to the static approach.

## 14.1 Introduction

Social networks are built by humans. And despite the fact that we are predictable to some extent, the social networks we build are in fact dynamic. The factors behind the dynamics may be of different nature, such as meeting new people, changing attitude towards others, switching the job, moving from one place to another and so on. Moreover, the intensity of contacts is also varying in time. In fact, the most accurate method of representing humans communication is the precise information about who contacted whom at which time. By having that it is possible to trace

R. Michalski (✉) · P. Kazienko
Department of Computational Intelligence, Wrocław University of Technology,
Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland
e-mail: radoslaw.michalski@pwr.edu.pl

P. Kazienko
e-mail: kazienko@pwr.edu.pl

D. Król et al. (eds.), *Propagation Phenomena in Real World Networks*,
Intelligent Systems Reference Library 85, DOI 10.1007/978-3-319-15916-4_14

the information or influence flow in social networks. Yet, without knowing about the content or the essence of the communication, only some assumptions about potential paths of information diffusion may be made. Moreover, as a single contact entry most often contains just two entities: sender and recipient (directed case) or contacting parties (undirected case), by using this level of granularity it is impossible to benefit from the mature apparatus of social network analysis (SNA), since no graph exists yet. It also requires a lot of storage comparing to another method—building a time-aggregated social network from the contact logs. In this case the information about communication gets aggregated and the intensity of contacts is most typically expressed as weights over edges (contacts) between nodes (individuals), as presented in [6]. This approach allows to obtain a broader view of interconnections in networks, distinguish groups, hubs, nodes on boundaries of the network and lets to perform other analyses that are offered by SNA techniques [13]. Unfortunately, while the time-aggregated view of the network is used, from an information or influence propagation point of view [43, 44], the most important aspect is missing: the order of contacts that is crucial in analysing the flow of information. As it was stated in [62], in these networks one assumes transitive paths, and this assumption does not hold in temporal or most granular representations of social networks. Moreover, as the contacts within social networks are often *bursty* [4], the static representation of networks will also ignore this fact leading to wrong conclusions about the dynamic processes taking place there. This is especially crucial when modelling the spread of epidemics, since the accuracy of predictions may strongly influence the potential actions in healthcare [53]. To not to loose the temporal information, researchers more and more often use temporal representation of networks and a comprehensive overview of methods of building temporal networks may be found in [34]. In this work the authors state that the literature on static graphs is many times larger than on temporal graphs and this is for a natural reason: it is much easier to analyse static graphs, especially analytically. Naturally, it is not a reason to avoid this direction, since as the research reveals, only temporal networks are representing the surrounding world accurately—the static approach is considered as leading to wrong conclusions about dynamical processes.

Looking from the perspective of social influence, this process has enough psychological and sociological complexity itself and because of that it shouldn't be analysed by using simplified underlying layer. It is modelled upon sociological assumptions about how people become influenced [32] and using time-aggregated networks to model humans' interactions definitely does not help in understanding the speed and directions of spread of influence. As it will be presented later in this chapter, most of traditional methods for analysing the influence processes in social networks base on a static representation, especially in the area of maximizing the spread of influence [39]. Since in this work the problem of influence maximization has been proven to be NP-hard, many heuristics were proposed, but mostly for time-aggregated networks. Nowadays the direction of research on modelling dynamic processes in networks should consider dynamic networks as a base, since by simplifying the reality the obtained results and drawn conclusions may be wrong.

The goal of this work is to present the state of the art in the area of maximizing the spread of influence in social networks, the limitations of it in the static networks and recent trends in using the dynamics of networks or past propagations to obtain better results. To achieve this, most common models of the social influence are presented in Sect. 14.2. Section 14.3 shows what are the variants of the challenges in social influence in networks, since the influence maximization is not the only one. Next, methods for maximizing the influence in static networks are introduced in Sect. 14.4. Then the concept of *Temporal Social Networks* is introduced and most typical representations of them are presented in Sect. 14.5. Having these introduced, Sect. 14.6 reveals the experimental study results showing that the temporal approach may outperform the static one for simple heuristics when considering the temporal underlying layer for the spread of influence. Section 14.7 comes back to presenting the state of the art, but presents how researchers try to take the advantage from the network dynamics or history to obtain better results in the challenge of maximizing the spread of influence.

## 14.2 Modelling the Spread of Influence

Before presenting the most popular models of social influence, it is worth to quote one sentence from [32] by Watts and Dodds:

> (...) it is still the case that formal models of social influence suffer from a dearth of realistic psychological assumptions.

The problem of fitting the real-world data to models and trying to answer the question whether particular influence processes may be modelled with a chosen approach is still challenging. It lies in the complexity of human behaviour and the impossibility of separating social processes that are occurring simultaneously. Still, many results achieved in this area tend to contradict this pessimistic point of view of Watts and Dodds and continuous development of models or models' variations suggests that models will fit the reality even better in next few years [2]. On the other hand, there still remains the gap between formal models and psychological explanation that requires to be intensively studied to find the psychological rationale of particular behaviour expressed in these models.

Since the strength of social influence depends on many factors such as the intensity of relationships between people in the networks, the network distance between users, temporal effects, characteristics of networks and individuals in the network [69], it is relatively hard to model all these factors combined. However, vast of research shows that under some assumptions there exist models that fit the reality well. Below the most important models that are most commonly used in this area are presented: *the Linear Threshold model*, *the Independent Cascade model*, *the Voter Model*, and *the Naming Game*. Each of them incorporates the sociological background of the influence process, but as was previously stated, sometimes it is just a loose interpretation of humans' behaviour, that, luckily, still fits the reality well for some cases. For these models their recent variants which are suitable for real-world scenarios are described as well.

From the historical perspective, studying the social influence in terms of analytical process was the case of trying to model how the influence spreads in time. Starting from a set of influenced nodes in time $t_0$ which are in this work denoted as $\Phi(0)$, as time unfolds, more and more of neighbours of $\Phi(0)$ become influenced if they fulfil the model criteria. Most typically, these processes are modelled in directed graphs and focus on a *progressive* case, where nodes may become *influenced* from *uninfluenced* state, not the other way round [39]. Since this is a network approach, the influence process occurs through edges in graph and most typically no other external factors of influence are considered, such as out-of-network sources.

### *14.2.1 The Linear Threshold Model*

The most recognizable model for social influence is Granovetter's Linear Threshold model [31]—*LT*, but similar approach was also proposed in [66]. In this model, a node $v$ is under influence of its influenced neighbours $w$ denoted as $N_v^{inf}$ according to a weight $b_{v,w}$, such that $\sum_{w \in N_v^{inf}} b_{v,w} \leq 1$. Each node $v$ has a *threshold* $\theta_v$ from the interval [0, 1] and this threshold represents the level that has to be met by the aggregated sum of $v$'s neighbours influence weights in order to influence the node $v$. So the formal condition of influencing the node $v$ is as follows:

$$\sum_{w \in N_v^{inf}} b_{v,w} \geq \theta_v. \tag{14.1}$$

The influence process ends where more nodes cannot be influenced—this is the formal stop condition for the static case. The way how this model works is presented in Fig. 14.1, while formally it is presented as Algorithm 14.1 for the case of uniformly assigned threshold values $\theta_v$ (based on [76]).

Here, the value of $\theta_v$ represents the individual's chances of becoming influenced when its neighbours are influenced. So all the psychological factors are included in this parameter and it should be also underlined that this approach represents the individual's perspective rather than the influencer perspective. Granovetter illustrated the model with the hypothetical case of a riot. Since individuals were unsure what are the costs and benefits of joining it, they observed their peers and considered joining only when sufficiently many of their neighbours joined the riot, otherwise they refrained.

Of course, the biggest question is how to assign particular values of $\theta$ to individual nodes and there are two most typical approaches: draw them from a probability distribution $f(\theta)$ as introduced in [31] or hard-wiring them at a fixed value [7, 61]. The most interesting and realistic scenario is the former one, i.e. drawing $\theta_v$ from a distribution, since the distribution represents both the average tendencies and also the heterogeneity present in the population. Lowering or raising the mean

**Algorithm 14.1** Linear Threshold model

```
Require: Graph $G(V, E)$, set of initially influenced nodes $\Phi(t_0)$
return Final set of influenced nodes $\Phi(K)$
k = 0;
Uniformly assign random thresholds $\theta_v$ from the interval [0, 1];
while $k = 0$ or $\Phi(t_{k-1}) \neq \Phi(t_k)$ do
    $\Phi(t_{k+1}) = \Phi(t_k)$;
    $uninfluenced = V \setminus \Phi(t_k)$;
    for all $v \in uninfluenced$ do
        if $\sum_{w \text{ influenced neighbour of } v} b_{v,w} \geq \theta_v$ then
            influence $v$;
            $\Phi(t_{k+1}) = \Phi(t_{k+1}) \cup \{v\}$;
        end if
    end for
    $k = k + 1$;
end while
$\Phi(K) = \Phi(k)$;
Return $\Phi(K)$;
```

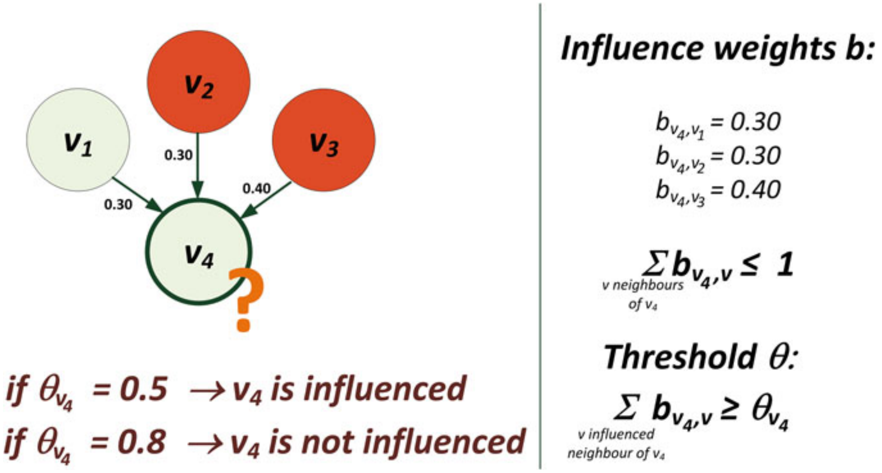

$v_2$
$v_1$
0.30
$v_3$
0.40
0.30
$v_4$
?
if $\theta_{v_4}$ = 0.5 → $v_4$ is influenced
if $\theta_{v_4}$ = 0.8 → $v_4$ is not influenced
Influence weights b:
$b_{v_4,v_1} = 0.30$
$b_{v_4,v_2} = 0.30$
$b_{v_4,v_3} = 0.40$
$\sum_{v \text{ neighbours of } v_4} b_{v_4,v} \leq 1$
Threshold θ:
$\sum_{v \text{ influenced neighbour of } v_4} b_{v_4,v} \geq \theta_{v_4}$

**Fig. 14.1** The illustration showing how the LT model works

of $f(\theta)$ would modify the general susceptibility of the population, while increasing or decreasing the variance would correspond to an increase or decrease in variability in susceptibility across individuals [32]. Still, hard-wired thresholds are also often considered in the research. An exemplary spread of influence process following the *LT* model is presented in Fig. 14.2.

It should be underlined that this process is time-independent, since it considers iterations rather than time. However, in most research works in this area an iteration represents a single time step, this is why the notation of $t_0, \ldots, t_K$ is often used instead of iterations $i_0, \ldots, i_K$.

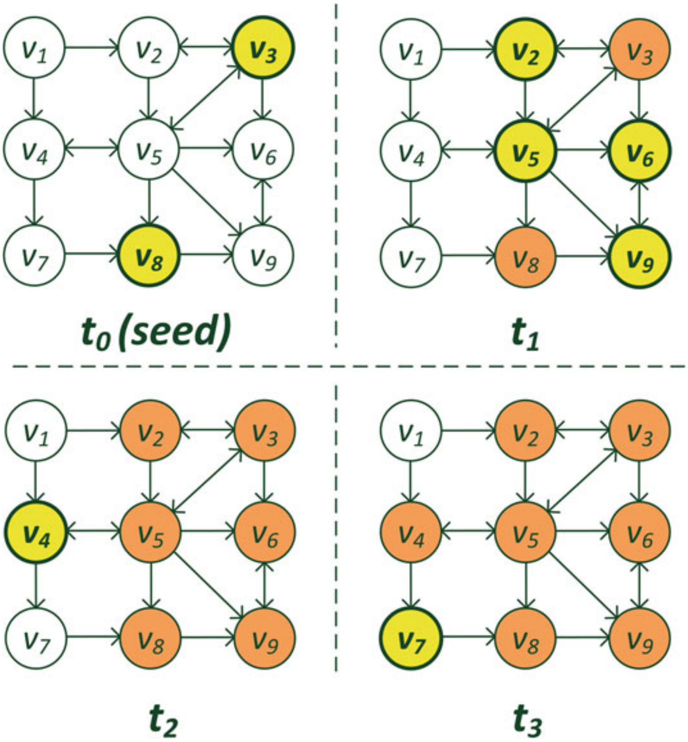


**Fig. 14.2** An exemplary social influence process following the linear threshold model in graph $G$. The threshold value is fixed for all nodes, $\theta_v = 0.33$. At the beginning $\Phi(0) = \{v_3, v_8\}$, at the end of the process $\Phi(t_3) = V(G) \setminus \{v_1\}$, since $v_1$ cannot be influenced for this model parameters. Nodes in *bold* were influenced at this particular process step

The following theorem and proof are excerpted from [39] and show the NP-hardness of influence maximization problem for the LT model. In the same work the proof for NP-hardness of the same problem for IC model is shown.

**Theorem 14.1** *The influence maximization problem is NP-hard for the Linear Threshold model.*

*Proof* Consider an instance of the NP-complete *Vertex Cover* problem defined by an undirected $n$-node graph $G = (V, E)$ and an integer $k$; it is expected to find a set $S$ of $k$ nodes in $G$ so that every edge has at least one endpoint in $S$. It is shown that this can be viewed as a special case of the influence maximization problem. Given an instance of the *Vertex Cover* problem involving a graph $G$, a corresponding instance of the influence maximization problem by directing all edges of $G$ in both directions is defined. If there is a vertex cover $S$ of size $k$ in $G$, then one can deterministically make $\sigma(A) = n$ by targeting the nodes in the set $A = S$; conversely, this is the only way to get a set $A$ with $\sigma(A) = n$. □

The LT model became a core of many modifications or extensions. For instance, [26] extends this model by introducing temporal decay, as well as factors such as the influence-ability of a specific user, and influence-proneness of a certain action. On the other hand, [5] proposes topic-aware extensions of *LT* model. In [60] the authors consider multiple cascades of *LT* model and they allow nodes to switch between them, whereas [10] introduces a number of modifications to the competing model variant: the authors force nodes to draw one cascade they join at the end of the process or consider the mutual influence of cascades on each other.

### 14.2.2 The Independent Cascade Model

The next model has its roots in interacting particle systems [23, 49] and is called the Independent Cascade model—*IC* [25, 39]. Again, the process starts with a set of influenced nodes $\Phi(0)$, but each node $v$ in the network has a probability $p_{v,w}$ assigned. According to this probability, the node $v$ gets a single chance to influence its neighbour $w$ and when it fails, it will have no other chance. If it succeeds, $w$ will become influenced in the next time step. Similarly to the LT model, the process runs until no more influences are possible.

From psychological perspective, in this model the influencer becomes more important, since he or she holds the probability $p_{v,w}$. This is one of the major differences between *IC* and *LT* models. In the *LT* model, the influence process parameter was assigned to the uninfluenced node and in the *IC* model it is hold by the potential influencer. Just as in the previous model, the probability may be fixed or drawn from a distribution $f(p)$. The Independent Cascade model is presented in Algorithm 14.2 (based on [76]).

**Algorithm 14.2** Independent Cascade model

```
Require: Graph $G(V, E)$, set of initially influenced nodes $\Phi(t_0)$, activation probabilities $p_{v,w}$
return Final set of influenced nodes $\Phi(K)$
k = 0;
while $\Phi(t_k) \neq \{\}$ do
    $k = k + 1$;
    $\Phi(t_k) = \{\}$
    for all $v \in \Phi(t_{k-1})$ do
        for all $w$ neighbour of $v$, $w \notin \cup_{j=0}^{k} \Phi(t_j)$ do
            $rand$ = generate a random number in [0, 1];
            if $rand < p_{v,w}$ then
                influence $w$;
                $\Phi(t_k) = \Phi(t_k) \cup \{w\}$;
            end if
        end for
    end for
end while
$\Phi(K) = \cup_{j=0}^{k} \Phi(t_j)$;
Return $\Phi(K)$;
```

Again, there are many variants of the *IC* model. Already mentioned work of [5] introduces the topic-aware approach also for this model, while [40] study the *decreasing cascade model*. One of the problems with the base *LT* and *IC* models is that they do not provide the influence probabilities and there are works that try to obtain them from past propagations. It may not be considered as a extension of a base model, but a way to make the probabilities or threshold more realistic. One of the works in this area is [65], but this topic will be covered in Sect. 14.7.1. There also exists the approach to model multiple independent cascades in the network [8].

### 14.2.3 The Voter Model and the Naming Game

An interesting case of influence in networks is the situation where two separate opinions or influences are competing in the society. This phenomenon may be observed in many situations and it has its roots in studying the consensus processes [51] or the language dynamics [20]. Below there are two variants of the process presented: the Voter Model (*VM*) and the Naming Game (*NG*).

The Voter Model introduced in [19] and extensively analysed later in [33] assumes that each node in the network can hold one of two opinions and by interacting with others it may switch the opinion to the opinion of the peer. This model introduces also the degree of conformity which defines whether a node will follow the majority (conformist) or minority (non-conformist), see [37].

On the other hand, the Naming Game, also referred to as binary-agreement model [74], introduces another variant of forming the opinion or spreading the influence. At any time a node may possess one of two competing opinions or two opinions simultaneously. In a given time step, we choose a node randomly, designate it as a speaker and choose one of its neighbours randomly and it is a listener. The speaker proceeds to convey its opinion to the listener (chosen randomly if it possesses two) to the listener. If the listener possesses this opinion already, both speaker and listener retain it while eliminating all other opinions; otherwise, the listener adds the opinion to his list [73].

Both of these models are useful in studying common phenomena occurring in social networks that involve binary options, such as reaching the consensus on contradictory opinions or observing which of competing parties will win the election. The current research trends suggest that these models will be actively studied and extended in the future [48, 52, 58, 64, 77].

### 14.2.4 Summary

The above presented models are just a selection of models that allow to analytically study the influence processes in social networks. As it was presented, they differ by the perspective (*LT* vs. *IC*), by the number of competing influences (*VS* and *NG* vs. others), but all of them are linked to the same process—spread of influence.

Sometimes their applicability is limited, but as far the empirical research shows, they model the human behaviour accurately in some cases, even if the psychological background of an individual is more complex than just a single parameter.

## 14.3 Social Influence Challenges

The work of Goyal et al. [28] brings some insights into the problem of influence in social networks. Since most often only the case of maximizing the spread of influence with a given budget $k$ was considered, there are some other research questions in this area. One should ask about minimizing the time of influence (number of iterations to influence a given number of nodes by having the budget $k$) or about minimizing the budget $k$ to influence a given set of nodes. The generalization of the challenges in this area may be considered as a constrained optimization problem, as presented in Fig. 14.3. The dimensions that can be optimized are the budget, the time and the number of influenced. Here, only one or two dimensions may be constrained and the third is optimized. Below, each of the dimensions is briefly described to show how they are understood by most of the researchers.

### *14.3.1 Budget*

The *budget* in the spread of influence problem in social networks is considered as an amount of the resource that can be spent on influencing nodes (please note that some literature prefers the more general term *activating*, e.g. see [39]). This resource is most often expressed as a budget $k$ of different nature such as money, gifts, conversations. However, each successful influence of a node in the network reduces the budget. Typically, it is assumed that the amount of a budget taken for influencing a node is equal for all of the nodes in network. Sometimes it may be

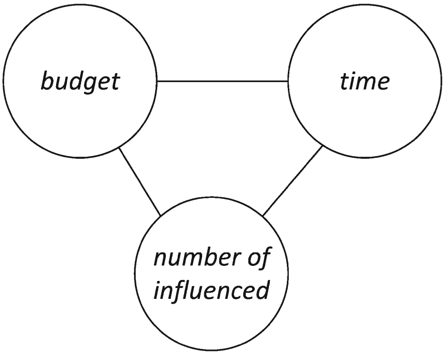


**Fig. 14.3** The optimization problem for the influence in social networks

true, e.g. if in a marketing campaign the same product is being sent to different customers the cost of distributing the product among them is considered to be equal. On the other hand, as the influence process is a subjective one, even by spending some amount of the budget on a user, he or she may not become influenced and this susceptibility may differ from user to user. As it was already presented, those psychological aspects are included in the spread of influence model properties (such as $\theta$ for the LT model or $p_{v,w}$ for the IC model), but these models do not consider the varying cost of influencing individuals in the set of seeds $\Phi(0)$. However, for the problem stated in this chapter, as well as in the research in this area, it is assumed that the cost of initial influencing a node is equal for all the nodes.

### *14.3.2 Time*

The time constraint expressed here means that we want to influence nodes in a given time and to evaluate the results of different methods this particular time is considered as a stop condition. Typically, the models work until no more nodes could be influenced. This is a natural stop condition and it is reasonable for static networks and described models. Of course, when the time constraint appears, the process may be evaluated sooner. But when considering the temporal social networks, the use of this hard stop condition may become more complicated, since if the network changes, the influence process may be infinite, e.g. if with every iteration new nodes join the network it could be hard to say at which moment the algorithms should stop. This is why the time constraint may be crucial for temporal social networks, since it introduces a moment which allows to compare methods.

From the marketers perspective, if they spent some budget on influencing nodes, they want to get the return from this investment in a given time, e.g. they want to have people interested in buying the product when it is offered and not discontinued [16]. On the other hand, other examples may be not so focused on the time dimension. For instance, spreading good manners among the society is one of the examples. Naturally, the sooner the habits will improve the better, but the time aspect is not so important here compared to the whole success of the campaign. This is why the use of time constraint is sometimes desired, but sometimes this dimension is being left unconstrained. However, reader should have in mind that for temporal social networks the use of time constraint is somehow natural, since the stop condition that no new nodes will become influenced may be wrong.

### *14.3.3 Number of Influenced*

The last, but most often considered dimension is the *number of influenced*. The number of influenced means how many nodes were influenced or activated in the process. From the historical perspective, this dimension was the one that was maximized while the other two left constrained or unconstrained, but [28] started to consider some other variations of the problem, e.g. by trying to minimize the time of influencing a given number of nodes.

However, when analysing advancements in research, the problem of maximization the influence is the most popular one among researchers.

## 14.4 Maximizing the Spread of Influence

The problem of finding most influential seeds in a social network was originally stated in [22]. The authors posed a question on how to pick nodes and *influence* them with some idea to maximize the overall spread of this idea across network. In this work the example of a marketing campaign was used and the researchers considered the network value of a customer, i.e. the benefits for the company if this customer will influence its neighbours. In fact, nowadays selling techniques often base on viral marketing, so it seems that the business strongly believes in a potential of such an approach. The influence of nodes on each other was modelled as a Markov random field [41] and obtained results revealed that this approach may be promising. In the next work on this topic the authors used a linear model where the solution for influence maximization based on solving linear equations [63]. However, what this model lacked for, was the iterativeness, since it reflected the joint distribution over all nodes. Compared with the psychological research on social influence or diffusion of innovation, the process is rather iterative, so the models representing it are most often of this kind.

### *14.4.1 The Greedy Algorithm*

The work that showed different approach to the one presented by Domingos and Richardson was [39]. The authors started by assuming that the influence is more an iterative process, so they analysed two models of this kind, namely LT and IC. By basing on these, firstly they considered the hardness of the influence maximization problem and in both cases it was proved to be NP-hard, see Theorem 14.1 with the corresponding proof for LT model and [39] for the proof for IC model. Then, by taking the advantage of the properties of submodularity [67] and the research on greedy hill-climbing algorithm they show that the greedy method may outperform classic approaches based on network measures, such as top degree or top betweenness. In fact the authors show that the outcomes of this approach may be not worst than 63 % of the optimal solution. The greedy method pseudo-code is presented as Algorithm 14.3 (based on [18]). Here, given a social network $G = (V, E)$ consisting of sets of vertices and edges, an initial seed set of $k$ nodes is being chosen iteratively which maximizes the influence. In each step of the algorithm a single vertex is chosen, such that the influence of the set $\Phi$ and this vertex is the greatest. Unfortunately, this algorithm has few drawbacks. One is the efficiency, since the influence is estimated with R simulation steps [18]. The other is that the algorithm is trying to pick nodes that maximize the influence in each iteration. When comparing it to the chess game, it always chooses the move that is giving the best position at the moment, not thinking about the next move. Sometimes it may be better to look at a combination of moves or nodes rather than a single next move to maximize the overall result and the greedy algorithm avoids it by its nature, since it finds the local optimum. However it still provides acceptable results comparing to the optimal solution, but sacrificing the efficiency.

**Algorithm 14.3** Greedy algorithm for maximizing the influence

```
initialize $\Phi(0) = \emptyset$ and $R = 10000$
for $i = 1$ to $k$ do
  for each vertex $v \in V \setminus \Phi(0)$ do
    $s_v = 0$
    for $i = 1$ to $R$ do
      $s_v = |Inf(\Phi(0) \cup \{v\})|$
    end for
    $s_v = s_v / R$
  end for
  $\Phi(0) = \Phi(0) \cup \{\arg\max_{v \in V \setminus \Phi(0)} \{s_v\}\}$
end for
output S
```

To overcome the drawbacks mentioned above, the research splits in two directions. Firstly, a number of techniques were proposed to optimize the greedy algorithm. Secondly, researchers started to search for new ways of maximizing the spread of influence. Below it is presented how the greedy algorithm was improved and later on new ideas on maximizing the influence in social networks are introduced.

### *14.4.2 Greedy Algorithm Optimization*

In the work of Leskovec et al. [46] there is also one more drawback of the greedy algorithm shown. As for now it was assumed that the cost of acquiring a single node is equal to others, but in social networks it may not be the case. For instance, for an ongoing marketing campaign its designers like to give to influential social network users some incentives to end up with higher spread of influence, their expectations for value of incentive may vary from one to another. On the other hand, there are some scenarios where the same products are being sent to different users assuming that they become influenced, so the potential cost is equal. However, since the influence process is a subjective rather than an objective one, the same gift may not result with the same satisfaction of users from the product. In contrast, there are some cases of social networks where the equal cost is possible. For instance, in epidemiology it is often assumed that cost or probability of infecting a person is the same for all the population, but this is not the case of influence. In sensor or computer networks also dealing with devices may introduce the same cost for each of them, but the assumption of the same cost for different users seems to be limited.

The mentioned work [46] analysed the case of different cost for influencing each node and proved that with this assumption the greedy algorithm performs badly. To overcome this limitation the authors introduce a novel approach, *Cost-Effective Forward selection* (*CEF*) that uses the greedy algorithm and the cost-sensitive method in parallel and the results of these methods are compared later to find the better one that will be used. Moreover, again by using the submodular properties of the cost function, the researchers are able to reduce the number of possible runs of evaluation of the quality of selected node ($Inf(\Phi \cup \{v\})$), because they base on the fact that the marginal increase of benefits with each added node does not increase more than in previous evaluation. This approach is called by them *Cost-Effective Lazy Forward selection* (*CELF*) and comparing to the greedy algorithm is up to 700 times faster with still acceptable results of at least $\frac{1}{2}(1-\frac{1}{e})$ of optimal solution. Comparing it with the greedy algorithm, which proposes $(1 - \frac{1}{e} - \varepsilon)$, makes CELF a very good rival.

However, at least for the IC model, there was still space for improvement, as shown in [18]. Here, the authors decided to base on random graphs in order to reduce the number of runs $R$ (see Algorithm 14.3). Their approach assumes that for IC model it is possible to reduce the graph of influences to only these edges that are potentially reachable from the set $\Phi$ at $i$th iteration. This change allows to gain additional 15–34 % improvement in running efficiency by keeping the same level of quality. More interestingly, in the same work the researchers propose new heuristic that significantly improves the influence spread while running more than six orders of magnitude faster than all greedy algorithms—*DegreeDiscountIC*. This approach is basing on a degree heuristic, but *discounts* the degree of a considered as a seed node $v$ by the value of already influenced neighbours of $v$, since there exists a non-zero probability that this node will become influenced by one of its influenced neighbours and it makes it less attractive as a seed.

Goyal et al. shown that further exploitation of submodularity may lead to even better results for greedy algorithms. In [29] there is an extension of CELF algorithm that leads to at least 35 % gain in performance. The idea is to store a heap for all non-selected nodes that contains the information not only about a marginal gain of particular node, but also the marginal gain of the best node from these evaluated before this node. Due to this trick there is no need to recalculate marginal gain of a node if this node was not selected resulting in less iterations of an algorithm.

### *14.4.3 Avoiding Greedy Search*

One of the approaches was already mentioned, it was the *DegreeDiscountIC* algorithm [18]. The same authors proposed another method [17], *Maximum Influence*

*Arborescence* (MIA), which also exploits the submodularity. However, in this case for the IC model, for each pair of nodes maximum influence paths are calculated. Then paths below a specified threshold of influence are discarded, focusing only on local regions of influence. Afterwards, these paths creating tree structures which do need to be updated often are joined and the calculation of the influence spread may be done recursively. This leads to significant gains in effectiveness of the algorithms comparing to others and introduces no loss in terms of quality. Moreover, the threshold parameter may be interpreted as a way of controlling the time of influence and the overall spread.

Another work [38] also avoids the greedy approach while outperforming it in terms of speed (2 to 3 orders of magnitude) and bringing similar results in accuracy. Authors claim that this is the first approach that uses simulated annealing [55] in solving the problem. In the case of influence maximization this approach starts with random seeds and then tries to move in the space of possible solutions (initial seeds) towards the local minimum by swapping at most one node in the seed set until the stop condition will be applied.

An interesting insight into the problem was given by Shakarian and Paulo in [68] where the authors propose an algorithm that guarantees to activate (influence) the whole network. The solution does not find the minimal set of seeds, but its outcomes may be compared to the budget $k$—if the seed set is less or equal $k$, the algorithm will fulfil the requirements and, moreover, the whole network will be influenced. The approach base on removing edges in the graph (by basing on the idea of shell decomposition [12]), but it also guarantees to influence the whole network.

The last mentioned algorithm in this section is *Simpath* that is intended to maximize the influence for the LT model [30]. This algorithm is operating on paths of influence in the social network by assuming that most of the influence is local. Results reveal that the algorithm outperformed the MIA method, considered as the state of the art in the task of influence maximization for static networks [17].

### *14.4.4 Summary*

All the above presented techniques may be considered as purely structural ones. In here, researchers do not use any kind of attributes of nodes other than their structural properties, such as location in the network or interconnectivity with other influenced nodes. The only parameter that may differentiate the nodes is the cost of influence, but in most cases it was assumed to be uniformly distributed. This approach of basing just on network structural properties makes the proposed algorithms universal, since they do not base on any network-specific attributes. As it will be shown later, there is also another emerging direction in this research that is basing on the data, so the merit of the social network communication. Moreover, it is worth emphasizing how many of the presented approaches took the advantage of the submodularity property.

However, all algorithms presented in this section suffer from one drawback which makes them just a rough simplifications of reality. Here, the network dynamics is not considered, and as it was already stated in Sect. 14.1, the ignorance of this fact may lead to wrong conclusions about the outcomes of the process. Before taking the reader in the area of research which tries to benefit from the dynamics of the network to maximize the influence, in the next section theoretical framework for representing dynamic networks is presented. Then a small empirical study in Sect. 14.6 shows that it is worth to make use from network dynamics, and following this direction current advancements in the topic of maximizing the spread of influence in the dynamic configuration are presented.

## 14.5 Temporal Social Networks

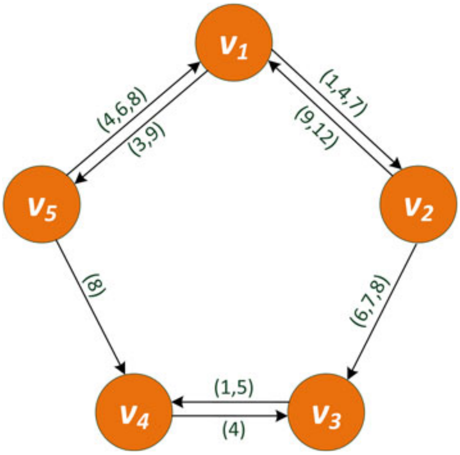
As it was already mentioned, real world social networks are rarely static. Our interactions occur in an ordered way, sometimes they are bursty, sometimes these contacts are suspended, but nevertheless there is a dynamics embedded. Until the era of IT the problem of gathering the data about interactions was definitely harder than nowadays, so it was another reason (but not the only one) why researchers based mostly on static networks. But when the era of electronic communication begun, it is relatively easy to track human communication, at least for some communication channels, such as e-mails, phone calls, instant messengers or social networking sites interactions. Moreover, there exist projects that try to obtain the data of real-world interactions, e.g. hospital ward contacts [71], conferences' participants [14] or students' behaviour [24]. Now the real research question is how to benefit from this time-annotated data to extend the knowledge on the social influence.

Before trying to find the answer on it, it is worth to discuss how these kind of data should be represented to not to loose the temporal information. One of the most extensive survey work on Temporal Social Networks *TSN* [34] depicts two major approaches in representing the temporal information in social networks, depending on the contacts type. These are enumerated and briefly described below.

### *14.5.1 Contact Sequences*

A *contact sequence* is obtained mostly from communication data where single contact between individuals is timestamped. As a relatively straightforward migration from event logs, it represents actors as nodes and edges between them have the temporal information, i.e. at which time a communication occurred. This approach is especially suitable when the duration of interactions is negligible, so it is preferred when representing asynchronous communication, such as e-mails, text messages and, to some extent, phone calls. The most typical formulation of contact sequences is as follows: for a set of vertices $V$ a contact sequence is a set of triples $(v_i, v_j, t)$ representing contacts between nodes $v_i$ and $v_j \in V$ at time $t$—see Fig. 14.4.

**Fig. 14.4** The representation of contact sequences. Here, for instance, node $v_1$ contacted $v_2$ in time *1*, *4*, *7* whereas the reverse communication occured in *9*, *12*



Very often this contact sequence is transformed into graphs where contacts in the same timeframe are grouped, making them a sequence of time-ordered static social networks [11], as presented below:

$$\begin{aligned} TSN^m &= \langle T_1, T_2, \ldots, T_m \rangle, && m \in \mathbb{N} \\ T_t &= SN_t(V_t, E_t), && t = 1, 2, \ldots, m \\ E_t &= \langle v_i, v_j \rangle : v_i, v_j \in V_t, && t = 1, 2, \ldots, m. \end{aligned} \tag{14.2}$$

In this formulation *TSN* represents the sequence of static networks $SN_t$ aggregating contacts in timeframe *t* making it more a evolving static social structure. However, the authors of [34] argue that this representation may miss many important points of temporal activity, what indeed is true, since some simplifications in the contact orders arise. On the other hand this simplification may be helpful in applying most popular models of social influence. The highest level of aggregation which results with a single static social network is often called a *time-aggregated graph*.

### 14.5.2 Interval Graphs

Another form of temporal networks are *interval graphs*. Here, in opposite to contact sequences, the edge is active in a period of time, rather than it appears at specific time. This kind of temporal networks is more suited for respecting the duration of contacts, so its application is also different than in the former approach. One of the examples might be tracking the duration of interpersonal contacts as the exposure on infection—the longer the exposure, the higher the probability of becoming infected. Here the edges are not active over a set of times but rather over a set of intervals $T_e = \{(t_1, t'_1), \ldots, (t_n, t'_n)\}$, where the parentheses mark the periods of activity.

The above representations of temporal networks offer the highest granularity which makes them perfect to track precise interactions between nodes. Naturally, depending on the research goal, some simplifications may be used, but it is worth to remember what consequences particular simplifications introduce. In the next subsection problem of transitivity in temporal networks is presented, since it is important in understanding the limitations of time-aggregated approach in analysing the diffusion or influence processes.

### 14.5.3 Limitations of the Time-Aggregated Approach

Consider the following graphs: $SN_{AGG}$ which consists of 4 nodes, namely $V_{AGG} = \{v_1, v_2, v_3, v_4\}$. This is a time-aggregated version of obtained contact sequences. Contact sequences were presented as graphs $SN_1, SN_2, SN_3$, named so, because contacts occurred in times $1, 2, 3$, respectively. For an illustration, see Fig. 14.5, where (a) denotes the time-aggregated graph and (b) contact sequences unfolded to three social networks.

Assuming that in the network a linear threshold process takes place with the threshold set uniformly for all nodes $\theta = 0.5$ and initially only node $v_4$ is influenced, the process will behave differently for both networks. For the time-aggregated graph after two iterations all nodes will become influenced. For the temporal network in $t = 1$ no new nodes will activate, the same in $t = 2$. In time $t = 3$ the node $v_4$ will be able to influence its neighbours and still $v_2$ will be not influenced. This simple example shows that the influence process is prone to the network dynamics and having this in mind the next section presents a simple experimental study showing that it is possible to benefit from it to obtain better influence maximization results.

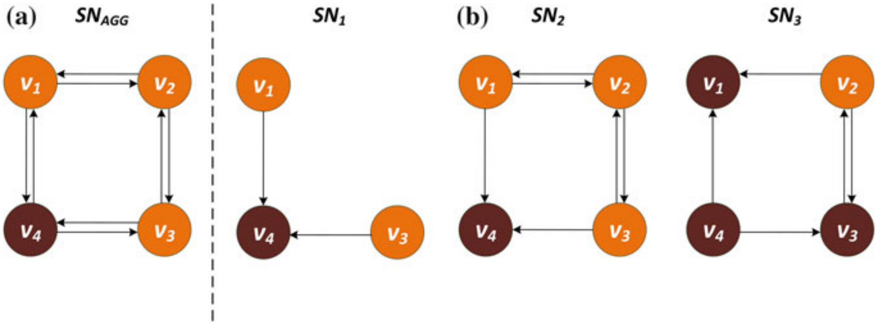


**Fig. 14.5** The LT influence process for time-aggregated graph and a temporal network

## 14.6 Spread of Influence—Temporal Approach

In the research which was presented in [56] it was decided to evaluate how the most typical heuristics based on the network structure perform on temporal and static networks. A special attention has been turned to observation of dynamics of the influence that spreads over temporal network after choosing initial seed sets. The basic heuristics basing on structural network measures were evaluated to see whether time-enhanced versions of these measures will perform better. Since the goal of this chapter is to present the current advancements in the research on maximizing the social influence, the experiments are briefly described here just to show that indeed moving with the spread of influence towards dynamic networks may be beneficial in solving the problem and it places it in real-world setup rather than in an abstract one.

### *14.6.1 Introduction*

The general problem considered in that context is what kind of networks should be used to perform better in seeding and finally in the spread outcome. Two main network kinds have been further studied: static one that aggregates equally all knowledge from the past ($TSN^1$ in Fig. 14.6) and temporal one that splits the past period into more or less time intervals: $TSN^{10}$ with 10 equal time windows and $TSN^5$ with 5 time frames. The temporal approach corresponds to dynamic context of seeding, whereas an aggregated social network reflects typical static seeding circumstances. The goal of the experiment was not to focus on seeding strategies itself, but to analyse the process under different assumptions for the aggregation level.

### *14.6.2 Experimental Setup*

#### 14.6.2.1 Time-Dependent Measures

Firstly, three simple aggregations were introduced, which allow to order users based on structural measures (total degree, in-degree, out-degree, betweenness, closeness) respecting all periods in the temporal social network in the accumulated way—maximum, minimum and sum. These aggregations, however, do not make use of sequential nature of time and general phenomena that recent social relationships are likely to be more influential than old ones. Hence, the nine new aggregations that take into account also the “forgetting” aspect of time are introduced. Here, the value of a given structural measure in the most recent time window is the most important, while the measures value in the oldest period is the least valuable. The purpose of this was not only to capture the dynamics of user behaviour but also to emphasize users

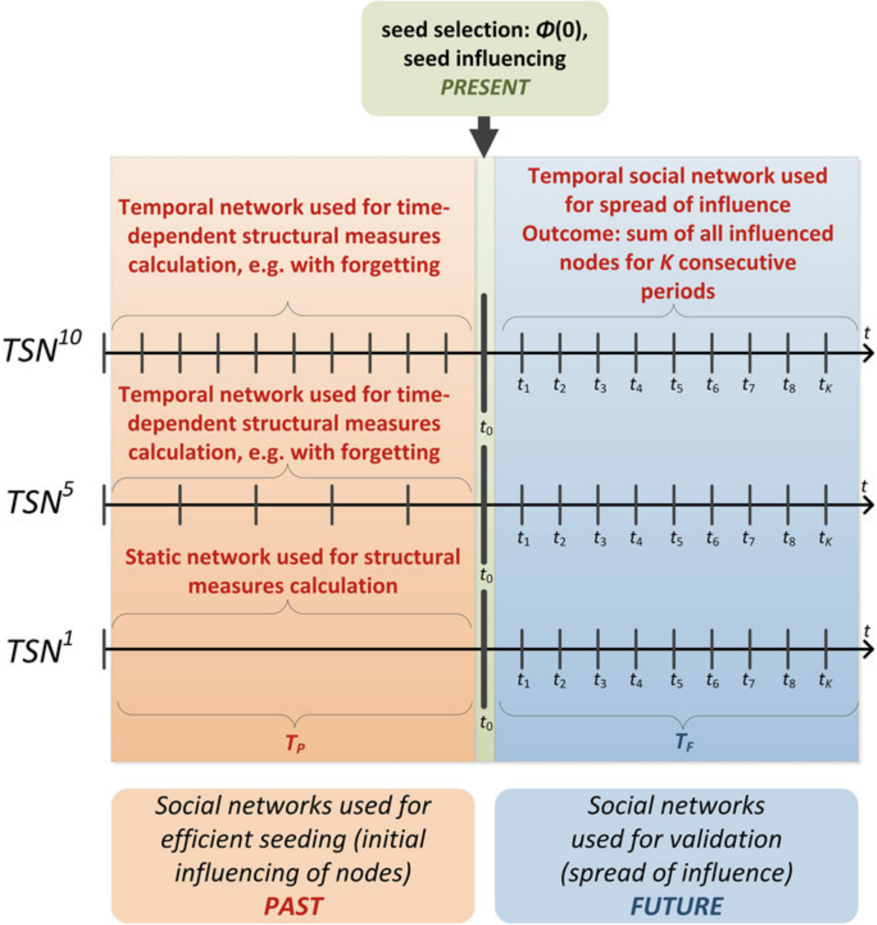

seed selection: Φ(0),
seed influencing
PRESENT
Temporal network used for time-dependent structural measures calculation, e.g. with forgetting
Temporal social network used for spread of influence
Outcome: sum of all influenced nodes for K consecutive periods
TSN10
t
t1 t2 t3 t4 t5 t6 t7 t8 tK
t0
Temporal network used for time-dependent structural measures calculation, e.g. with forgetting
TSN5
t
t1 t2 t3 t4 t5 t6 t7 t8 tK
t0
Static network used for structural measures calculation
TSN1
t
t1 t2 t3 t4 t5 t6 t7 t8 tK
t0
TP
TF
Social networks used for efficient seeding (initial influencing of nodes)
PAST
Social networks used for validation (spread of influence)
FUTURE

**Fig. 14.6** Seeding is performed at present based on the knowledge about past dynamics of the social network (in time $T_P$). The seed—set $\Phi(0)$ of initially influenced nodes is used to spread of influence in the dynamic social network in the future (in time $T_F$). Three kinds of 'learning' social networks used in the experiments on seed selection are depicted one below another: $TSN^{10}$ with 10 time windows, $TSN^5$ with 5 time frames, $TSN^1$—aggregated-static (one time window)

latest activities. So the new aggregations were applying different kinds of forgetting, e.g. linear, hyperbolic or exponential forgetting.

### 14.6.2.2 Aggregation Levels and Influence Model

All the aggregations combined with all typical node structural measures (in-degree, out-degree, total degree, betweenness and closeness) where used to create node rankings and select the seed set for spreading the influence. In other words, nodes in the temporal social network from the past were ranked according to the time-aggregated values of their structural measures and this aggregation was performed for all component networks used for seeding, see the left part of Fig. 14.6. Next, 5 % of top ranked nodes were used for seeding, see the middle part of Fig. 14.6. It means that these top nodes form the initial set $\Phi(0)$ of already influenced nodes that may influence others in the following periods, see the right part of Fig. 14.6. In each case, the second part of the dataset was split into ten windows of equal duration, to reflect the dynamic behaviour of the network. As a model of influence, the linear threshold was used and three levels of $\theta$ were evaluated—0.33, 0.50, 0.75 assigned uniformly for all nodes.

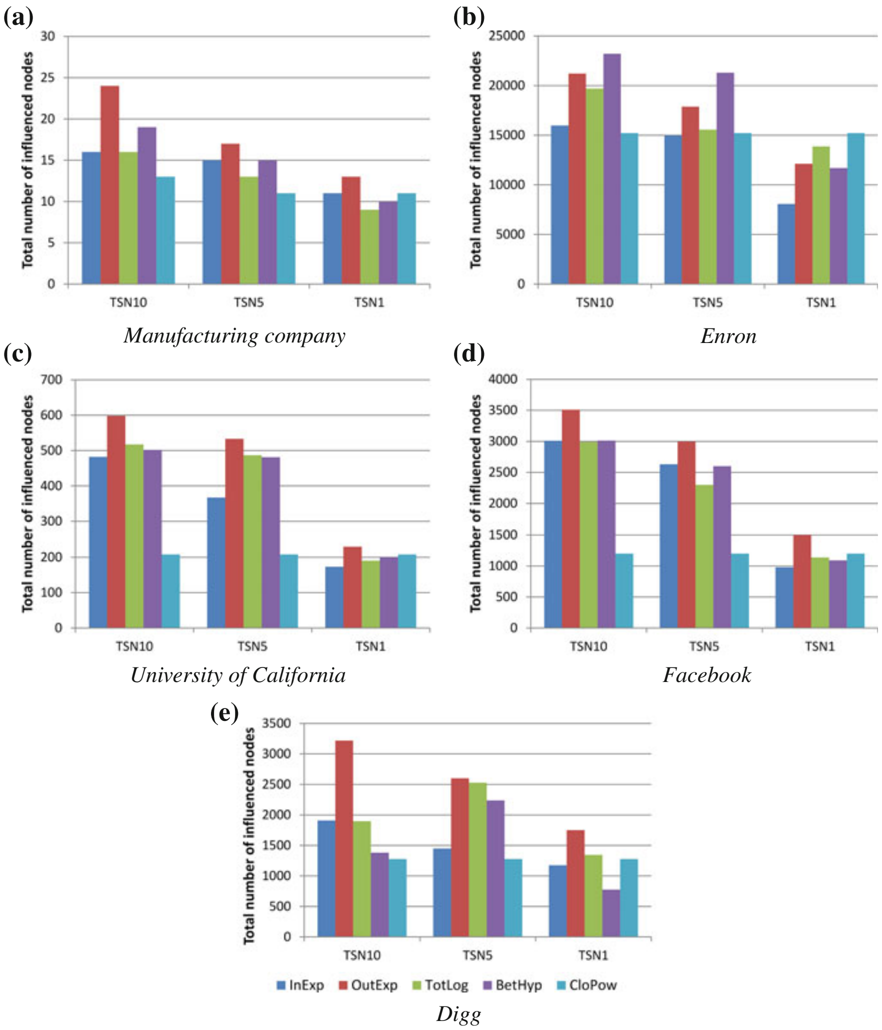
#### 14.6.2.3 Datasets

The experiments were conducted using five real-world social networks representing the communication between company employees or social services users (Table 14.1). All of them were extracted from communication datasets downloaded from the Koblenz Network Collection (KONECT)[1] repository. Each social network has timestamped edges, so it allowed to perform temporal analysis. The properties of the datasets are presented in Table 14.1.

**Table 14.1** Descriptions and basic properties of used datasets

| Dataset ID | Network description | No. of nodes | No. of timestamped edges | Period of communication |
|---|---|---|---|---|
| 1 | E-mail communication between employees of manufacturing company [57] | 167 | 82,927 | 2010-01-02 ... 2010-09-30 |
| 2 | The Enron email network [42] | 87,101 | 1,147,126 | 1998-11-02 ... 2002-07-12 |
| 3 | Messages sent between the users of an online community of students from the University of California, Irvine [59] | 1,899 | 59,835 | 2004-04-15 ... 2004-10-26 |
| 4 | Facebook user to user wall posts [72] | 46,952 | 876,993 | 2004-09-14 ... 2009-01-22 |
| 5 | The reply network of the social news website Digg [21] | 30,398 | 87,627 | 2008-10-28 ... 2008-11-13 |

[1] http://konect.uni-koblenz.de.



*Manufacturing company*

*Enron*

*University of California*

*Facebook*

*Digg*

**Fig. 14.7** The total number of influenced nodes for all networks and structural measures used for seeding as well as for different datasets, the threshold level $\theta = 0.75$. **a** Manufacturing company. **b** Enron **c** University of California **d** Facebook **e** Digg

## *14.6.3 Results*

Results revealed that indeed for the aggregated (static) network, i.e. $TSN^1$, the total number of the influenced nodes is the lowest (the right group of bars in Fig. 14.7) and the best performing network type is the one with the biggest number of time windows, i.e. $TSN^{10}$—the left hand side group of bars, Fig. 14.7. Overall, the final number of influenced nodes for the 10-windows networks ($TSN^{10}$) was about double as much as for a single network *TSN*[1], see Fig. 14.7. In this figure the first part of the heuristics name is the measure type, i.e. In—in-degree, Out—out-degree, Tot—total degree, Bet—betweenness, Clo—closeness. The second part of the name represents the type of forgetting—exponential (Exp), logarithmic (Log), hyperbolic (Hyp) or power (Pow), for details see [56].

It confirms the initial hypothesis that using dynamic network it is possible to better utilize the information in original data and finally select better seeds. When about introduced measures, the ones based on forgetting properties outperformed others.

What is more, the greater granularity, the better chance to choose the proper seeds, especially if taking time into consideration by means of time-dependent measures, such as based on linear forgetting. When trying to explain this phenomenon, once again the intuition is suggesting that the increasing granularity is helpful in terms of better representation of the network dynamics, so the sensitivity of the introduced measures increases—they reflect dynamics to a greater extent.

### *14.6.4 Summary*

The above presented experiment shows that indeed the temporal aspects of networks are helpful in building seed strategies and that this direction should be further exploited to verify the initially confirmed assumption that the network dynamics helps in maximizing the spread of influence. In the next section it is shown how researchers try to make use of this direction by presenting recent advancements. As the literature overview shows, the problem here is being solved in different ways, not only as in the presented experiment, but all of these have something in common—they look at the history to maximize the spread.

## 14.7 Maximizing the Spread of Influence in Dynamic Networks

This section is devoted to presenting recent advancements in maximizing the spread of influence in dynamic networks which are definitely closer to real-world setting. Since most of the research presented here makes the approaches more data-dependent, reader has to have in mind that in contrast to purely structural algorithms described in Sect. 14.4, the application of the approaches introduced below may be limited, since not always the researchers have the full information about a social networks (e.g. communication content). Of course it is not an argument to avoid this direction, but just a loose remark.

### *14.7.1 Learning Influence Probabilities*

Before thinking about maximizing the spread in real-world dynamic networks, there should be at least one limitation overcome, which is how to assign the influence probabilities making them more aligned to the reality. As it was presented in Sect. 14.2, the influence is often drawn from a distribution or hard-wired. When thinking of real-world scenarios, this assumption makes the results of modelling spread of influence questionable. In this area there are just few papers which try to deal with this problem.

The research on this topic started with the work of Tang et al. [70]. Here, the researchers try to avoid learning influence probabilities from the network position of a node, since they assume that different peers of a node may have different influence on it. For instance, our friends may be more influential in the area of private live (trends, friends etc.), while relatives from work may have stronger influence in company-related topics. To take this into account, the authors decided to analyse the content of the communication to build a model of *Topical Affinity Propagation* (TAP). This approach tries to assign influence values over edges between nodes which are topic-specific. So in this case a node may have multiple edges with its neighbour and each of them represents different topic altogether with different influence weights. The authors base on a concept of *factor graphs* [45], in which the observation data are cohesive on both local attributes and relationships. Moreover, to make the approach scalable, they do the following: define a *Topical Factor Graph* (TFG), then they introduce *Topical Affinity Propagation* and finally they try to make the approach scalable for large networks, either by using Map-Reduce approach or a parallel update rule. Their main goal of the proposed idea is expert identification, but this approach suits well for just learning real influence probabilities, as the real-datasets experiments show.

Another approach in this area was proposed by Saito et al. in [65]. Here, the authors focus on the IC model and they base on a likelihood of so-called *episodes*, which are in fact nodes that became influenced in consecutive time-windows. Then they compare neighbouring episodes—the one in time $t$, which is $D(t)$ in authors' notation, and the next one in $t+1$ (defined as $D(t+1)$) to see whether the neighbouring nodes were in $D(t)$ and $D(t+1)$. If so, it is probable that the newly influenced node $v_z$ from $D(t+1)$ was influenced by $v_y$ from $D(t)$ in time $t+1$. It is possible, because the IC model gives just a single chance to a node to influence its neighbours, so the influence may happen only shortly after the node itself becomes influenced. Then the researchers use the expectation maximization technique to obtain the values of likelihood functions of $\theta$. Experiments conducted on a blogging platform confirm that this approach may be right in terms of obtaining the influence probabilities by learning from past data.

The last work presented here is the work of Goyal et al. [26]. In contrast to the previous work the authors generalize their approach to every influence model following the submodularity property, which makes it more universal (e.g. covering LT and IC). As the reader may remember, the submodularity was the property which allowed to introduce many improvements in the area of maximizing the spread of influence, see Sect. 14.4. In this work the researchers base on two sources: the temporal network and an action log which represents the activity of users. In detail, the action log is defined as a relation containing tuples $(u, a, t_u)$, where $u$ represents a node $\in V$, $a$—an action from *the universe of actions* and $t_u$ the time when the user $u$ performed the action $a$. By proposing models for capturing static and dynamic influence the authors are able to compute the probabilities of influence and they reduce the number of scans of typically-huge action log. Moreover, they are able to predict at which time a user will take an action.

### *14.7.2 Real-World Datasets Evaluation*

An important set of hints of how to seek for influential individuals is obtained from the analysis of real-world datasets. In here the case is not to learn influence probabilities to be applied for artificial models, but to get the understanding of what really matters by analysing the influence paths. This information is helpful in building real-world strategies and supplements the mathematical approaches by the knowledge on how influence works in variety of social networks. The only drawback of such analysis is that it is mostly data-dependent, since it emphasizes attributes of users or networks which cannot be generalized easily.

One of the most interesting works in this area is [15]. Authors analyse Twitter social network in order to find the most important factors of individuals that have the greatest influence. Results revealed that the in-degree measure is not a good influence indicator, at least for Twitter. Users with high in-degree not necessarily are the ones that also have the significant influence on others. Moreover an interesting conclusion is drawn that individuals are influential across many topics, i.e. the influence is rarely topic-dependent, but node-dependent. Lastly it was shown that nodes cannot build their influence instantly, it is rather a long-lasting process of becoming important in the neighbourhood. This conclusion suggests that it is barely impossible to insert into the network individuals that become influential fast. Instead the marketing strategies should focus on finding influential nodes and convincing them to opt for a product or service. Some other research on looking for influential nodes on Twitter is presented in [3, 75].

The next study uses Epinions.com portal dataset to find out who is responsible for the most influential reviews of products. By using text-mining techniques the influential power of real online users through their reviews was calculated and combined with the RFM solution which tracks users past behaviour [35]. It was observed that users contributing the most are not necessarily perceived as influential ones, i.e. the experience based on the number of written reviews is not the most important. Secondly, the most important reviewers wrote the reviews in a very emotional manner, i.e. they put a lot of effort to make them sound very subjective instead of objective. It means that neutral reviews are not considered as valuable by others.

It is observed that the approach of finding influential nodes corresponds well to obtaining the influence probabilities, since in the former strategy it is possible to adjust close-to-real influence probabilities, while the latter shows which factors may be additionally used as the ones for choosing the seed set.

### *14.7.3 Social Influence Maximization*

The approach which combines the action log and a social graph presented by Goyal et al. in [26] was later extended to maximize the spread of influence in [27]. Firstly, the authors show that basing on real-world data (action logs or history of past propagations) is crucial, since only by knowing real influence probabilities any algorithm for influence maximization may be accurate. So the initial assumption is that these probabilities have to be computed by real-world data. Then they propose so-called *Credit Distribution model* (CD) which bases on different assumptions than LT and IC models, since it considers actions as a source of influence in network. Authors introduce propagation graphs which include nodes that were neighbours in graph $E$ and performed the same action but in different time. Here, a node performing an action earlier may be considered as a potential influencer of its neighbours taking this action later. So, in fact, the initial graph is static, but the actions introduce the dynamics here. Under the credit model for each action performed by a node, all nodes that took this action earlier and are neighboured to this node receive credits for being potential influencers, and this is a recursive operation. Then the nodes that maximize the influence in the whole network under so-defined model offering $(1 - \frac{1}{e})$—approximation comparing to the optimal solution are being found keeping the scalability as well. The biggest achievement here is the lack of need to perform costly Monte Carlo simulations, but it is because of different model definition. However, the results show that the CD model and the method to choose seeds allow to outperform common approaches for LT and IC influence maximization offering also speed improvement. It is worth to study this paper also because the authors compare the seed sets provided by different models.

In a work of Mathioudakis et al. [54] the researchers use past propagation log and a social graph to find $k$ most influential links, i.e. links that will maximize the propagation. However, what they do is making significant reduction in the search space by benefiting from sparsification. They apply their approach to IC model and propose a *Spine*, dynamic programming algorithm which propose a significant improvement in speed offering accuracy close to optimal. Spine is structured in two phases. During the first phase it selects a set of arcs $D_0$ that yields a log-likelihood larger than $-\infty$. This is done by means of a greedy approximation algorithm for the Hitting Set NP-hard problem. During the second phase, it greedily seeks a solution of maximum log-likelihood, i.e. at each step the arc that offers the largest increase in log-likelihood is added to the solution set [9].

An interesting approach of considering time-varying influences is proposed in [50] where authors consider the delayed influence process, i.e. the influence of a node to its neighbours may vary in time. It places the problem closer to reality, where people take more care to recent incidents rather than to the older ones. The authors propose *Influence Spreading Paths* as a method of measuring the influence of a node, i.e. $ISP(u, S)$ represents all spreading paths that end with user $u$. By using them the authors compute the activation probability of a user $u$ and thanks to that they are able to find seeds faster than by using greedy algorithm. So the time factor incorporated here is not the time reflecting the dynamics of the network but the changes in influence probabilities. However, the researchers of this chapter indicate that this is another way of representing the network dynamics and as such it can be used for solving the problem in a dynamic environment.

The problem of influence maximization in dynamic social networks was just recently explicitly stated in [1]. As the authors claim, to their knowledge they propose the first set for time-sensitive methods for influence maximization. In this work researchers use the transmission matrix which contains the time-dependent functions for influence spread to find solution for two separate problems. Firstly, they would like to pick $k$ nodes at time $t_1$ to maximize the influence at time $t_2$—this problem lies closer to classic influence maximization problem, but it incorporates the time factor. Secondly, when observing the influence spread at time $t_2$ they would like to know which nodes most probably were responsible for the influence spread at time $t_1$. To deal with these problems, they introduce *Backward and Forward Influence Algorithms*. When looking at influence maximization problem, authors try to solve it similarly to the greedy algorithm, but now each iteration means another time-step. In conducted experiments it is shown that the time-dependent solutions outperformed the static ones showing that this direction should be further exploited.

To two more works in this area which tackle the problem differently are worth referring. In [47] researchers try to find successor nodes for removed seeds. It is a relatively different research question than in a typical influence maximization problem, since now the budget $k$ will increase, but this approach incorporates the dynamics of networks showing that considering it is crucial. In [36] authors try to take into account the availability factor of nodes, which indeed is the embedded dynamics of the network, trying to improve the overall influence spread in networks. Again, experiments' results confirm that this direction helps the seeding strategies in obtaining better results.

The idea of maximizing spread of influence in dynamic networks is a relatively new one, but as the above literature review shows, it seems that considering the dynamics of networks in the influence process is important and already some solutions are being proposed. However, the work in this area has just begun and we should expect some improvements shortly. One of the reasons is that the dynamics in social networks is something natural rather than unusual and it is already agreed that the influence maximization problem should be considered in this real-world setup.

## 14.8 Summary

In this work the problem of influence maximization in social networks was presented. By starting with the most typical models of social influence, namely the Linear Threshold, Independent Cascade, Voter Model and the Naming Game the influence maximization for social networks was defined. After that it was shown how the solutions for it developed for the static case. However, as the static case barely fits the complex reality of dynamic social networks which are definitely more often found in real-world, a short introduction to temporal networks was presented just after. To confirm that the temporal solutions may be helpful in seeding strategies, a short experimental study was quoted and discussed. In the last part of this chapter the recent developments in the area of maximizing the spread of influence in dynamic networks were shown.

However, the problem lies in how to put all the information provided here into a synthetic form to conduct a successful marketing campaign. It seems that apart from theoretical approaches a great deal of real-world dataset evaluation suggestions has to be included in order to find out who is considered as an influential person in particular social network. If there is only a pure network structure available, designers of campaign have no other option then use purely structural approaches. But when attributes of individuals or time-sensitive data are obtained, the seed selection process should use dataset-dependent information. An interesting case would be a comparison of how data-dependent seeding strategies perform against structural ones. Yet it seems that ignoring the temporal information may be one of the worst strategies.

The idea of this work was to show how the problem developed and was solved in a chronological order and what are the current challenges in this area. Since the apparatus for temporal networks is already established [34] and the problem for the dynamic case is clearly stated [1], nothing should stop other researchers in this area from developing new algorithms for the real-world scenarios.

**Acknowledgments** This work was partially supported by the fellowship co-financed by the European Union within the European Social Fund, by the European Commission under the 7th Framework Programme, Coordination and Support Action, Grant Agreement Number 316097, ENGINE—European research centre of Network intelliGence for INnovation Enhancement (http://engine.pwr.wroc.pl) and by The National Science Centre, the decision no. DEC-2013/09/B/ST6/02317. The calculations were carried out in Wroclaw Centre for Networking and Supercomputing (http://www.wcss.wroc.pl), grant No 177.

## References

1. Aggarwal, C.C., Lin, S., Philip, S.Y.: On influential node discovery in dynamic social networks. In: SDM, pp. 636–647. SIAM, Anaheim (2012)
2. Aral, S., Walker, D.: Identifying influential and susceptible members of social networks. Science **337**(6092), 337–341 (2012)

3. Bakshy, E., Hofman, J.M., Mason, W.A., Watts, D.J.: Everyone's an influencer: quantifying influence on twitter. In: Proceedings of the fourth ACM international conference on Web search and data mining, pp. 65–74. ACM (2011)
4. Barabási, A.L.: Bursts: The Hidden Patterns Behind Everything we do, From Your E-mail to Bloody Crusades. Penguin (2010)
5. Barbieri, N., Bonchi, F., Manco, G.: Topic-aware social influence propagation models. Knowl. Inf. Syst. **37**(3), 555–584 (2013)
6. Barrat, A., Barthelemy, M., Pastor-Satorras, R., Vespignani, A.: The architecture of complex weighted networks. Proc. Natl. Acad. Sci. U.S.A. **101**(11), 3747–3752 (2004)
7. Berger, E.: Dynamic monopolies of constant size. J. Comb. Theor. Ser. B **83**(2), 191–200 (2001)
8. Bharathi, S., Kempe, D., Salek, M.: Competitive influence maximization in social networks. In: Internet and Network Economics, pp. 306–311. Springer, Heidelberg (2007)
9. Bonchi, F.: Influence propagation in social networks: a data mining perspective. IEEE Intell. Inf. Bull. **12**(1), 8–16 (2011)
10. Borodin, A., Filmus, Y., Oren, J.: Threshold models for competitive influence in social networks. In: Internet and Network Economics, pp. 539–550. Springer, Heidelberg (2010)
11. Bródka, P., Saganowski, S., Kazienko, P.: Ged: the method for group evolution discovery in social networks. Soc. Netw. Anal. Min. **3**(1), 1–14 (2013)
12. Carmi, S., Havlin, S., Kirkpatrick, S., Shavitt, Y., Shir, E.: A model of internet topology using k-shell decomposition. Proc. Natl. Acad. Sci. **104**(27), 11150–11154 (2007)
13. Carrington, P.J., Scott, J., Wasserman, S.: Models and Methods in Social Network Analysis. Cambridge University Press, Cambridge (2005)
14. Cattuto, C., Van den Broeck, W., Barrat, A., Colizza, V., Pinton, J.F., Vespignani, A.: Dynamics of person-to-person interactions from distributed RFID sensor networks. PloS ONE **5**(7), e11596 (2010)
15. Cha, M., Haddadi, H., Benevenuto, F., Gummadi, P.K.: Measuring user influence in twitter: the million follower fallacy. ICWSM **10**, 10–17 (2010)
16. Chen, W., Lu, W., Zhang, N.: Time-critical influence maximization in social networks with time-delayed diffusion process. arXiv:1204.3074 (2012)
17. Chen, W., Wang, C., Wang, Y.: Scalable influence maximization for prevalent viral marketing in large-scale social networks. In: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1029–1038. ACM (2010)
18. Chen, W., Wang, Y., Yang, S.: Efficient influence maximization in social networks. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 199–208. ACM (2009)
19. Clifford, P., Sudbury, A.: A model for spatial conflict. Biometrika **60**(3), 581–588 (1973)
20. DallAsta, L., Baronchelli, A., Barrat, A., Loreto, V.: Nonequilibrium dynamics of language games on complex networks. Phys. Rev. E **74**(3), 036105 (2006)
21. De Choudhury, M., Sundaram, H., John, A., Seligmann, D.D.: Social synchrony: predicting mimicry of user actions in online social media. In: Computational Science and Engineering, 2009. CSE'09. International Conference on, vol. 4, pp. 151–158. IEEE (2009)
22. Domingos, P., Richardson, M.: Mining the network value of customers. In: Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 57–66. ACM (2001)
23. Durrett, R., Durrett, R., Durrett, R., Durrett, R.: Lecture notes on particle systems and percolation. Wadsworth & Brooks/Cole Advanced Books & Software, Pacific Grove, CA (1988)
24. Eagle, N., Pentland, A.: Reality mining: sensing complex social systems. Pers. Ubiquitous Comput. **10**(4), 255–268 (2006)
25. Goldenberg, J., Libai, B., Muller, E.: Talk of the network: a complex systems look at the underlying process of word-of-mouth. Mark. Lett. **12**(3), 211–223 (2001)
26. Goyal, A., Bonchi, F., Lakshmanan, L.V.: Learning influence probabilities in social networks. In: Proceedings of the Third ACM International Conference on Web Search and Data Mining, pp. 241–250. ACM (2010)

27. Goyal, A., Bonchi, F., Lakshmanan, L.V.: A data-based approach to social influence maximization. Proc. VLDB Endowment **5**(1), 73–84 (2011)
28. Goyal, A., Bonchi, F., Lakshmanan, L.V., Venkatasubramanian, S.: On minimizing budget and time in influence propagation over social networks. Soc. Netw. Anal. Min. **3**(2), 179–192 (2013)
29. Goyal, A., Lu, W., Lakshmanan, L.V.: Celf++: optimizing the greedy algorithm for influence maximization in social networks. In: Proceedings of the 20th International Conference Companion on World wide web, pp. 47–48. ACM (2011)
30. Goyal, A., Lu, W., Lakshmanan, L.V.: Simpath: an efficient algorithm for influence maximization under the linear threshold model. In: Data Mining (ICDM), 2011 IEEE 11th International Conference on, pp. 211–220. IEEE (2011)
31. Granovetter, M.: Threshold models of collective behavior. Am. J. Sociol. **83**(6), 1420 (1978)
32. Hedström, P., Bearman, P.: The Oxford Handbook of Analytical Sociology. Oxford University Press, Oxford (2009)
33. Holley, R.A., Liggett, T.M.: Ergodic theorems for weakly interacting infinite systems and the voter model. The Annals of Probability, pp. 643–663 (1975)
34. Holme, P., Saramäki, J.: Temporal networks. Phys. Rep. **519**(3), 97–125 (2012)
35. Hughes, A.M.: Strategic Database Marketing. McGraw-Hill, New York (2006)
36. Jankowski, J., Michalski, R., Kazienko, P.: Compensatory seeding in networks with varying avaliability of nodes. In: Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, pp. 1242–1249. ACM (2013)
37. Javarone, M.A.: Social influences in the voter model: the role of conformity. arXiv preprint arXiv:1401.0839 (2014)
38. Jiang, Q., Song, G., Cong, G., Wang, Y., Si, W., Xie, K.: Simulated annealing based influence maximization in social networks. In: AAAI (2011)
39. Kempe, D., Kleinberg, J., Tardos, É.: Maximizing the spread of influence through a social network. In: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 137–146. ACM (2003)
40. Kempe, D., Kleinberg, J., Tardos, É.: Influential nodes in a diffusion model for social networks. In: Automata, Languages and Programming, pp. 1127–1138. Springer, Heidelberg (2005)
41. Kindermann, R., Snell, J.L., et al.: Markov Random Fields and their Applications, vol. 1. American Mathematical Society Providence, R.I. (1980)
42. Klimt, B., Yang, Y.: The enron corpus: a new dataset for email classification research. In: Machine learning: ECML 2004, pp. 217–226. Springer, Heidelberg (2004)
43. Król, D.: On modelling social propagation phenomenon. In: N. Nguyen, B. Attachoo, B. Trawiński, K. Somboonviwat (eds.) Intelligent Information and Database Systems, Lecture Notes in Computer Science, vol. 8398, pp. 227–236. Springer, Heidelberg (2014)
44. Król, D.: Propagation phenomenon in complex networks: theory and practice. New Gener. Comput. **32**(3–4), 187–192 (2014)
45. Kschischang, F.R., Frey, B.J., Loeliger, H.A.: Factor graphs and the sum-product algorithm. Inf. Theor. IEEE Trans. **47**(2), 498–519 (2001)
46. Leskovec, J., Krause, A., Guestrin, C., Faloutsos, C., VanBriesen, J., Glance, N.: Cost-effective outbreak detection in networks. In: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 420–429. ACM (2007)
47. Li, C.T., Hsieh, H.P., Lin, S.D., Shan, M.K.: Finding influential seed successors in social networks. In: Proceedings of the 21st International Conference Companion on World Wide Web, pp. 557–558. ACM (2012)
48. Li, Y., Chen, W., Wang, Y., Zhang, Z.L.: Influence diffusion dynamics and influence maximization in social networks with friend and foe relationships. In: Proceedings of the Sixth ACM International Conference on Web Search and Data Mining, pp. 657–666. ACM (2013)
49. Liggett, T.M.: Interacting Particle Systems. Springer, Berlin (1985)
50. Liu, B., Cong, G., Xu, D., Zeng, Y.: Time constrained influence maximization in social networks. In: ICDM, pp. 439–448 (2012)

51. Lu, Q., Korniss, G., Szymanski, B.K.: The naming game in social networks: community formation and consensus engineering. J. Econ. Interac. Coord. **4**(2), 221–235 (2009)
52. Maity, S.K., Mukherjee, A., Tria, F., Loreto, V.: Emergence of fast agreement in an overhearing population: the case of the naming game. EPL (Europhysics Letters) **101**(6), 68,004 (2013)
53. Masuda, N., Holme, P.: Predicting and controlling infectious disease epidemics using temporal networks. F1000Prime Reports 5, 6 (2013)
54. Mathioudakis, M., Bonchi, F., Castillo, C., Gionis, A., Ukkonen, A.: Sparsification of influence networks. In: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 529–537. ACM (2011)
55. Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., Teller, E.: Equation of state calculations by fast computing machines. J. Chem. Phys. **21**, 1087–1092 (1953)
56. Michalski, R., Kajdanowicz, T., Bródka, P., Kazienko, P.: Seed selection for spread of influence in social networks: Temporal versus static approach. New Gener. Comput. (2014) (in press)
57. Michalski, R., Palus, S., Kazienko, P.: Matching organizational structure and social network extracted from email communication. In: Business Information Systems, pp. 197–206. Springer, Berlin (2011)
58. Mobilia, M.: Commitment versus persuasion in the three-party constrained voter model. J. Stat. Phys. **151**(1–2), 69–91 (2013)
59. Opsahl, T., Panzarasa, P.: Clustering in weighted networks. Soc. Netw. **31**(2), 155–163 (2009)
60. Pathak, N., Banerjee, A., Srivastava, J.: A generalized linear threshold model for multiple cascades. In: 2010 IEEE 10th International Conference on, Data Mining (ICDM), pp. 965–970. IEEE (2010)
61. Peleg, D.: Local majority voting, small coalitions and controlling monopolies in graphs: a review. In: Proceedings of 3rd Colloquium on Structural Information and Communication Complexity, pp. 152–169 (1997)
62. Pfitzner, R., Scholtes, I., Garas, A., Tessone, C.J., Schweitzer, F.: Betweenness preference: quantifying correlations in the topological dynamics of temporal networks. Phys. Rev. Lett. **110**(19), 198,701 (2013)
63. Richardson, M., Domingos, P.: Mining knowledge-sharing sites for viral marketing. In: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 61–70. ACM (2002)
64. Rogers, T., Gross, T.: Consensus time and conformity in the adaptive voter model. Phys. Rev. E **88**(3), 030,102 (2013)
65. Saito, K., Nakano, R., Kimura, M.: Prediction of information diffusion probabilities for independent cascade model. In: Knowledge-Based Intelligent Information and Engineering Systems, pp. 67–75. Springer, Berlin (2008)
66. Schelling, T.: Micromotives and Macrobehavior. WW Norton and Company, New York (1978)
67. Schrijver, A.: Combinatorial Optimization: Polyhedra and Efficiency, vol. 24. Springer, Berlin (2003)
68. Shakarian, P., Paulo, D.: Large social networks can be targeted for viral marketing with small seed sets. In: Proceedings of the 2012 International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2012), pp. 1–8. IEEE Computer Society, Canada (2012)
69. Sun, J., Tang, J.: A survey of models and algorithms for social influence analysis. In: Social Network Data Analytics, pp. 177–214. Springer, New York (2011)
70. Tang, J., Sun, J., Wang, C., Yang, Z.: Social influence analysis in large-scale networks. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 807–816. ACM (2009)
71. Vanhems, P., Barrat, A., Cattuto, C., Pinton, J.F., Khanafer, N., Régis, C., Kim, B.a., Comte, B., Voirin, N.: Estimating potential infection transmission routes in hospital wards using wearable proximity sensors. PloS ONE **8**(9), e73970 (2013)
72. Viswanath, B., Mislove, A., Cha, M., Gummadi, K.P.: On the evolution of user interaction in facebook. In: Proceedings of the 2nd ACM Workshop on Online Social Networks, pp. 37–42. ACM (2009)

73. Xie, J., Emenheiser, J., Kirby, M., Sreenivasan, S., Szymanski, B.K., Korniss, G.: Evolution of opinions on social networks in the presence of competing committed groups. PloS ONE **7**(3), e33215 (2012)
74. Xie, J., Sreenivasan, S., Korniss, G., Zhang, W., Lim, C., Szymanski, B.K.: Social consensus through the influence of committed minorities. Phys. Rev. E **84**(1), 011,130 (2011)
75. Ye, S., Wu, S.F.: Measuring Message Propagation and Social Influence on Twitter.com. Springer, Heidelberg (2010)
76. Zafarani, R., Abbasi, M.A., Liu, H.: Social Media Mining: An Introduction. Cambridge University Press, Cambridge (2014)
77. Zhang, W., Lim, C., Korniss, G., Szymanski, B.: Spatial Propagation of Opinion Dynamics: Naming Game on Random Geographic Graph. arXiv:1401.0115 (2013)

# Index

D. Król et al. (eds.), *Propagation Phenomena in Real World Networks*,
Intelligent Systems Reference Library 85, DOI 10.1007/978-3-319-15916-4