University of Exeter

Department of Computer Science

# Community Detection in Complex Networks

Amenah Dahim Abbood Al-Dayyeni

August, 2018

Supervised by Professor Richard Everson & Professor Jonathan Fieldsend

Submitted by Amenah Dahim Abbood Al-dayyeni to the University of Exeter as a thesis for the degree of Doctor of Philosophy in Computer Science , August, 2018.



I certify that all material in this thesis which is not my own work has been identified and that no material has previously been submitted and approved for the award of a degree by this or any other University.

(signature) .....

# Abstract

Finding communities of connected individuals in social networks is essential for understanding our society and interactions within the network. Recently attention has turned to analyse these communities in complex network systems. In this thesis, we study three challenges. Firstly, analysing and evaluating the robustness of new and existing score functions as these functions are used to assess the community structure for a given network. Secondly, unfolding community structures in static social networks. Finally, detecting the dynamics of communities that change over time. The score functions are evaluated on different community structures. The behaviour of these functions is studied by migrating nodes randomly from their community to a random community in a given true partition until all nodes will be migrated far from their communities. Then Multi-Objective Evolutionary Algorithm Based Community Detection in Social Networks (MOEA-CD) is used to capture the intuition of community identification with dense connections within the community and sparse with others. This algorithm redirects the design of objective functions according to the nodes' relations within community and with other communities. This new model includes two new contradictory objectives, the first is to maximise the internal neighbours for each node within a community and the second is to minimise the maximum external links for each node within a community with respect to its internal neighbours. Both of these objectives are optimised simultaneously to find a set of estimated Pareto-optimal solutions where each solution corresponds to a network partition.

Moreover, we propose a new local heuristic search, namely, the Neighbour Node Centrality (NNC) strategy which is combined with the proposed model to improve the performance of MOEA-CD to find a local optimal solution.

We also design an algorithm which produces community structures that evolve over time. Recognising that there may be many possible community structures that explain the observed social network at each time step, in contrast to existing methods, which generally treat this as a coupled optimisation problem, we formulate the problem in a Hidden Markov Model framework, which allows the most likely sequence of communities to be found using the Viterbi algorithm where there are many candidate community structures which are generated using Multi-Objective Evolutionary Algorithm.

To demonstrate that our study is effective, it is evaluated on synthetic and real-life dynamic networks and it is used to discover the changing Twitter communities of MPs preceding the Brexit referendum.

## Acknowledgements

This a good opportunity to thank everyone who help me and contributed in realising this thesis specifically my supervisor, Prof. Richard Everson without him the thesis would never have been possible to finish. He always encourages me to do the best to get more experience throughout my PhD. I thank him for his time and patience.

I would like to thank forever my family who supported me in all aspects of my life. I am forever grateful to my husband Ammar Al-shahrablee who is the source of my strength, and he always stands by my side.

I would like to thank my second supervisor Prof. Jonathan Fieldsend for his guidance and time to finish this thesis. I would like to thank Dr Christopher Ferro, my mentor throughout my PhD. We are grateful to Iain Weaver, Hywel Williams, Iulia Cioroianu, Travis Coan and Susan Banducci for the MP Twitter data.

I also would like to thank the academic and technical staff of Exeter University for their support. I would like to thanks all my teacher in my life. I am also thankful Prof. Barra Ali Attea and Dr.Loay E. Geoge, Alma Rahat, Ali Kareem, Mohamed Younis, Huthaifa abo Hammad,Luma Harbi, Yuan Zuo, George De Ath, Chengqiang Huang for their friendship and support.

Special and deepest thanks express to the staff of Baghdad University, the university where I am employed, for nominating me to this PhD scholarship, which is provided by the Ministry of Higher Education of Iraq scholarship, the completion of this thesis would not have been possible without their grateful supports.

# Contents

# List of Tables

# List of Figures

# Nomenclature and Abbreviations

**Acronyms and abbreviations**

| | |
|---|---|
| CF | Community Fitness *(p. 17)* |
| CON | CONductance *(p. 4)* |
| CS | Community Score *(p. 4)* |
| EA | Evolutionary Algorithm *(p. 7)* |
| GA | Genetic Algorithm *(p. 20)* |
| ID | Internal Density *(p. 18)* |
| KKM | Kernel K-Mean *(p. 17)* |
| MOEA | Multi-Objective Evolutionary Algorithm *(p. 8)* |
| MOP | Multi-Objective Optimisation Problem *(p. 20)* |
| NC | Normalized Cut *(p. 4)* |
| NMI | Normalised Mutual Information *(p. 35)* |
| NNC | Neighbour Node Centrality *(p. 42)* |
| PS | Pareto Set *(p. 20)* |
| Q | Modularity *(p. 4)* |
| QD | Modularity density *(p. 39)* |
| RC | Ratio Cut *(p. 18)* |
| SC | Snapshot Cost *(p. 37)* |

TC Temporal Cost *(p. 37)*

# Publications

**The materials presented in chapter 4 have been published in:**

Dahim, Amenah and Everson, Richard, (2017). Detecting Dynamic Communities in Complex Networks Using Viterbi and Evolutionary Algorithms. In *Proceedings of the 2017 Conference on Complex Networks and Their Applications*, Complex Networks 6 '14, pages 344–347, Lyon, France, November 29 - December 01.

# Chapter 1

# Introduction

The recent science of network systems has brought important advances to represent and deeper understanding of the main characteristic of complex networks [Duan et al., 2014]. Understanding the configuration of networks provides positive effects in different fields such as computer science, engineering, biology, economics, etc. These techniques have attracted many researchers since these systems include real-world networks such as technological networks, networks, information networks (World Wide Web networks) [Broder et al., 2000], biological networks (protein-protein networks and neural networks) [Girvan and Newman, 2002], scientific collaboration networks [Newman, 2001], and transportation networks [Banavar et al., 2000].

A graph representation is the simplest method to represent complex real-world networks where nodes (vertices) represent objects such as individuals, neurones, proteins or countries and the connections between these objects are represented by edges or links such as communication, friendship or collaboration (for example, two authors (nodes) write a paper together and can therefore be joined by an edge). In general, a network can be divided into groups of nodes, called communities, modules or clusters, each group has dense connections within it (intra-connections) and is only sparsely connected with the rest (inter-connections), see Figure. 1.1. This type

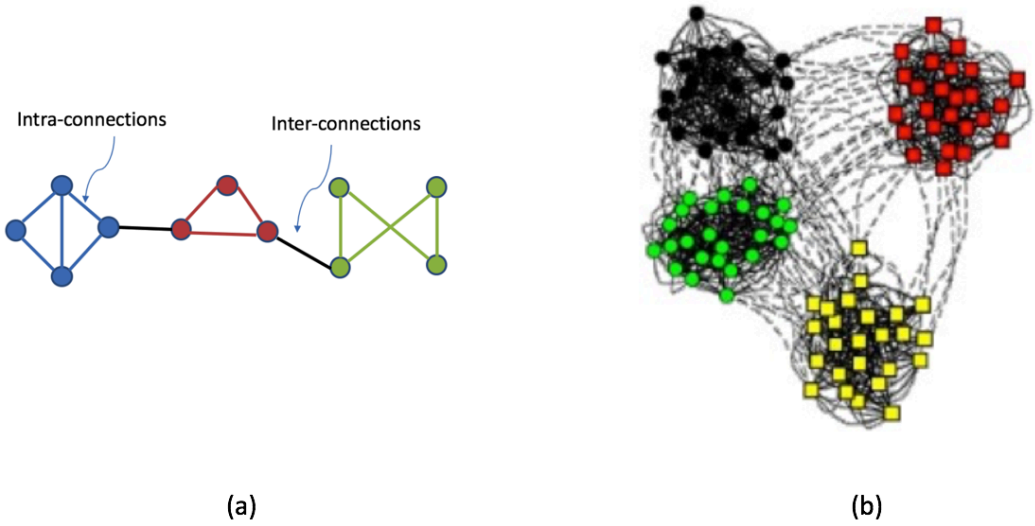


Figure 1.1 **(a) A graph is partitioned into three communities. (b) The synthetic network consisting of four communities [Lancichinetti et al., 2008]. Different colours for each community.**

of organisation is known as the graph's community structure. It is crucial to identify communities as it helps to determine the structural properties of the networks.

A common way of discovering community structure is to define an objective function or quality measure that quantifies the quality of a proposed partition of the nodes of a network into communities and then to optimise the objective function over possible partitions. This approach depends on (a) defining an appropriate quality function and (b) a search algorithm to efficiently locate good quality partitions. Generally speaking, all the proposed quality functions have the same two goals: they favour partitions with more connections inside communities and few connections with the rest [Girvan and Newman, 2002; Lancichinetti et al., 2010]. This definition of community is known and accepted by most network scientists. Also, it is similar to general clustering techniques [Jain and Dubes, 1988] which attempt to partition a set of data to maximise the similarity of members within the cluster and minimise similarity with the rest.

Many algorithms have been suggested to uncover communities in networks based on connections between nodes and these connections may be weighted or unweighted, directed or undirected [Girvan and Newman, 2002; Lancichinetti et al., 2011; Miyauchi

and Kawase, 2016; Pizzuti, 2012]. However, understanding these networks is very challenging especially when the configuration of these networks is evolving as many real-world networks are complex and dynamic. Therefore, analysing community structures is still an open problem and it needs more investigations. In 2004, Girven and Newman made a most exciting contribution to the community detection research area when they used the modularity score to measure the strength of partitioning of the network into communities [Newman and Girvan, 2004]. If the modularity is high, the network partition has dense connections between nodes within the community but sparse connections between different communities. Although it is the most popular measure for evaluating community structures, it suffers from a resolution limitation as it cannot detect small communities even if there are sparse connections between these small communities [Chen et al., 2014; Lancichinetti and Fortunato, 2011].

In the last few years, single objective evolutionary algorithms have been used to optimise a partition quality measure to detect community structure. Although these algorithms have been successful in identifying correct partitions on some real-world networks, they have failed on others because they have produced a solution with fixed property for the community structure as SOEA optimise one objective [Hafez et al., 2014]. Based on the general definition of communities, we can usefully consider the optimisation of the two objectives. One of them tends to increase intra-connections, while the second decreases inter-connections. In this way, this problem has been formulated as the multi-objective optimisation problem to produce a set of trade-off solutions by optimising two conflicting objective functions [Shi et al., 2012; Pizzuti, 2012; Gong et al., 2014; Wu and Pan, 2015].

Furthermore, Shi et al. [2014] proposed an algorithm to select objective functions in multi-objective community detection and they concluded that optimising two conflicting objectives using an evolutionary algorithm outperformed single objective evolutionary algorithms.

There is another important factor which should be taken into account when nodes and their connections change over time. It is generally expected that networks evolve slowly, so dramatic changes in community structure are unlikely. This is commonly modelled by the addition of a temporal cost that penalises abrupt changes in community structure from one time to the next. Existing methods for analysing community evolution have used only one objective to evaluate snapshot quality such as Modularity, Community Score, CONductance, Normalised Cut. [Folino and Pizzuti, 2010, 2014; Ma et al., 2014; Zhou et al., 2015] while the community detection issue is proved as multi-objective optimisation since networks having multiple structural properties [Shi et al., 2012; Pizzuti, 2012; Gong et al., 2014; Wu and Pan, 2015].

After this scenario, we wish to develop the analysis of the structure of communities in different types of synthetic and real-world networks. This analysis will be described in details in chapter 3 and 4.

The following chapters will present the algorithms that have been used to unfold community structures in static and dynamic networks.

## 1.1 Key Challenges and Novel Contributions

In the last few years, community detection has become an important research topic in complex network analysis. So far, many methods have been proposed to analyse these communities and provide a good understanding of the configuration of these networks, see [Fortunato and Hric, 2016] for a review. However, some of the critical issues are still open questions. In this section, we describe some of these issues and attempt to deal with them as our main contributions in this thesis.

### Objective Evaluation

As we discussed earlier, the nodes in the network are to be grouped by the community detection algorithm. In order to detect the structures of these communities, a score

function is needed to evaluate these structures. The study of score functions to determine the quality of a partitioning of the graph into communities is important. A few works of literature have been developed to evaluate the utility of the quality scores for community detection. Also, the evaluation of these scores based on the correct partition is missing in the literature. For example, Hafez et al. employed the single objective optimisation technique to investigate the quality of different objective functions [Hafez et al., 2014]. However, this method produces just one solution to validate the quality of the optimised objective with respect to the true partition. Therefore, the investigation in this research area is still not clear to determine the utility of scoring functions as the objective evaluation strategies need much of useful network partitions for the evaluation.

**A New Methodology for Evaluating Partition Quality Scores**

An evaluation technique is proposed to evaluate the accuracy of the community detection scores that are used to assess a given network partition. The proposed method is based on a random migration strategy and allows a proper empirical assessment of the suitability of objective functions for finding partitions.

**Community Detection in Static Networks**

Many quality measures have been suggested so far to uncover communities in networks [Girvan and Newman, 2002; Lancichinetti et al., 2011; Miyauchi and Kawase, 2016; Pizzuti, 2012]. Although these measures have been used to successfully detect the true partition or more similar to the true partition on some real-world networks such as the Karate and Dolphin networks [Zachary, 1977; Lusseau, 2003], they have failed on others such as American football network [Girvan and Newman, 2002]. The vast majority of the current score functions attempt to minimise all connections between communities, while in real networks the best partitions still have a few connections remaining between communities. This means that current score functions are ineffective at detecting some community structures in these networks.

Therefore, the critical question about the accurate structure of communities is still open: what score function or functions should be used to accurately recover the community structure in a wide range of real networks? The mathematical design includes the formulation of the intuition of intra-connections within a community and inter-connections with nodes outside the community. This issue needs more investigation and development on a variety of configurations of synthetic and real-world networks. An effective method is needed to capture the intuition of natural community identification.

### A New Multi-Objective Algorithm for Static Community Detection

We define two novel objectives for optimisation by an evolutionary algorithm. These objectives are optimised to find a set of network partitions that trade-off between intra-connections and inter-connections to reflect different network partitions in a single run. These objectives are inspired by our investigation of relations between nodes in the network rather than relations between communities which has characterised most previous work such as Modularity [Newman and Girvan, 2004], Community Fitness [Lancichinetti et al., 2009], Normalised Cut [Dhillon et al., 2004], etc. The first objective attempts to increase the number of connections for each node within the community with respect to external connections. In contrast, the second objective minimises the maximum connections between communities. The new algorithm is shown to successfully locate network partitions in synthetic and real-world networks (see chapter 3).

### Evolutionary Algorithms and Local Minima

Although the existing evolutionary algorithms for revealing community structure are able to explore large parts of the space of partitions, evolutionary algorithms tend to become stuck in local minima. In addition, few studies to tackle this issue by combining local search techniques with evolutionary algorithms for community detection [Gong et al., 2014; Wu and Pan, 2015; Hariz et al., 2016]. Therefore this

area needs more investigation to derive heuristic techniques by studying the natural network partition.

### Heuristic to Guide Evolutionary Algorithm Optimisation

We introduce a novel mutation operator that is based on neighbour relationships to assign nodes to more suitable communities, thus enhancing the performance of the mutation process and speeding up the convergence of the proposed evolutionary algorithm. In combination with the new objective functions, this heuristic facilitates the efficient location of community structure in complex networks.

### Community Detection in Dynamic Networks

Although many techniques are now available to computationally analyse unchanging networks, particularly for detecting communities of interconnected nodes (e.g., [Newman and Leicht, 2007; Hofman and Wiggins, 2008; Hafez et al., 2014; Wu and Pan, 2015; Chen et al., 2014]), a characteristic of many real networks is that they evolve over time [Hopcroft et al., 2004; Nguyen et al., 2014; Sun and Sun, 2017; Pizzuti, 2012; Gong et al., 2014] as nodes and edges are added or deleted. Identifying the changing structures of communities in these networks is important for understanding the underlying processes generating the networks and for making predictions about future configurations. However, analysing and understanding these structures is a challenging research topic because communities must be detected and tracked over time. Although several algorithms have been proposed to detect communities and their evolution in dynamic networks [Lin et al., 2009; Folino and Pizzuti, 2010; Tang et al., 2008; Xu et al., 2014; Ma et al., 2014; Cazabet and Amblard, 2014], further work is needed due to the wide variety of networks and how they evolve.

**Evaluating and Tracking the Structure of Communities in Dynamic Networks**

The detection of dynamic communities is formulated as a Hidden Markov Model to capture the evolution of these communities over time in the dynamic networks. Our Multi-Objective Evolutionary Algorithm (MOEA) is used to produce a candidate states at each time steps. Then the Viterbi algorithm is used to find the most likely sequence of network partitions over time as the communities evolve in time. We demonstrate the efficiency of the proposed algorithm on synthetic and real networks.

## 1.2 Organization of the Thesis

The thesis is organised as follows.

**Chapter 2**

We present background and the relevant work for analysing communities in static and dynamic networks. In particular, we discuss the existing score functions for evaluating a given network partition. Then, we review community detection algorithms that have been used for detecting of community structures in static and dynamic networks and discuss their advantages and disadvantages.

**Chapter 3**

In this chapter, we evaluate the utility of existing and novel objective/score functions to identify the correct network partition. We then formulate two novel objective functions that characterise the intra and inter-community connections in a network. These are incorporated into an MOEA algorithm for community detection in static networks. This algorithm is combined with a heuristic strategy to speed up the convergence of our algorithm. We evaluate our model against three existing models with and without a local heuristic search, on synthetic and real-world networks.

**Chapter 4**

We examine the detection of communities evolving over time. This problem is formulated as a Hidden Markov Model (HMM) to find the most likely network partitions over time. The MOEA developed in chapter 3 is used to generate a set of possible partitions (states) at each time step by optimising two objective functions. A Viterbi algorithm is then used to find the most likely sequence of partitions. The proposed algorithm is evaluated on synthetic and real-world networks, and the evolving social network structure between MPs in the approach to the Brexit referendum is analysed.

**Chapter 5**

This chapter summarises our contribution for evaluating and detecting community structures in static and dynamic networks and the results that are presented in this thesis. In addition, we present the possible directions that are derived from our study for future work.

# Chapter 2

# Background

In this chapter, we introduce some fundamental concepts of graph theory and relevant studies which have a relation to the aim of this thesis. Specifically, we discuss the current evaluation scores that have been used to judge whether the generated network partition is fit to a given network or not. Following this, we present the current literature on detecting community structure in both static networks where the given network is a snapshot at a specific time and dynamic networks when the input is a set of snapshots at successive time steps.

## 2.1 Network Theory

Network theory is the research area concerned with the analysis and understanding of the structure of complex networks. It is a part of graph theory, which is a mathematical method for modelling relations between objects (the structure of the graph). We begin by describing the basic notations that will be used throughout this thesis and some ways of characterising the structure of a network.

We model a static network as a graph $\mathcal{G} = (V, E)$, where $V$ represents the set of nodes or vertices, $V(\mathcal{G}) = \{v_1, v_2, \ldots, v_N\}$ with $N = |V|$ and $E(\mathcal{G})$ represents a

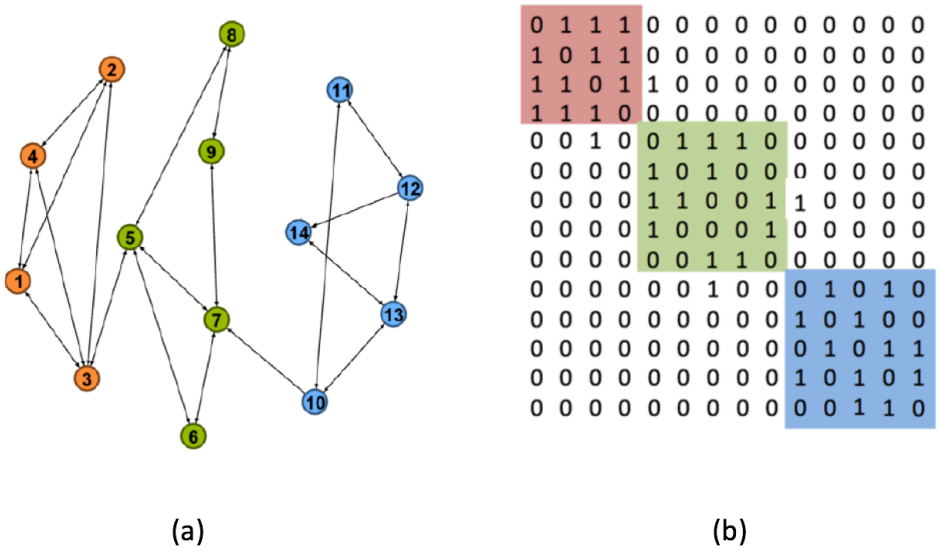


Figure 2.1 (a) A graph which consists of three communities. (b) Adjacency matrix which consists of three communities.

set of $L$ links or edges between nodes; $L = |E(\mathcal{G})|$. The graph is considered to be undirected and unweighted. Each node has some connections to other nodes, and this number of connections is the degree $d$ of the node. Let $\mathcal{G}$ be represented as an adjacency matrix $A \in \mathbb{R}^{N \times N}$, where $A_{ij} = 1$ if there is a link (edge) between $v_i$ and $v_j$ where $i, j \in \{1, 2, ..., N\}$, while $A_{ij} = 0$ otherwise. The adjacency matrix contains all the important information about the graph. Each row and column is indexed by a node's number, and each element indicates whether there is a link between a pair of nodes or not. All elements on the main diagonal in the adjacency matrix are zero as there are no connections between a node and itself. Figure 2.1a shows a graph that partitions into three communities in different colours and Figure 2.1b displays the corresponding adjacency matrix for the graph representation that partitions into three communities shown in three different colours.

The objective of community detection is to partition the graph, or equivalently, $A$ into a set of $K$ clusters or communities $\mathcal{C} = \{C_1, C_2, \ldots, C_K\}$. We denote the number of nodes in cluster $C_k$ as $n_k = |C_k|$.

**Degree**

As noted above the degree of node $v_i$ is the number of edges between node $v_i$ and other nodes:

$$d(v_i) = \sum_{j=1}^{N} A_{ij} \tag{2.1}$$

In addition to the degree of a node, we consider the degree of communities. The number of links within a particular community and between the community and other communities is important for the definition of communities. The *degree* of community $C_k$ is defined as:

$$D(C_k) = \sum_{i \in C_k} \sum_{j=1}^{N} A_{ij} \tag{2.2}$$

We also define the *external degree*,

$$\overline{D}(C_k) = \sum_{i \in C_k} \sum_{j \notin C} A_{ij} \tag{2.3}$$

and the *internal degree*

$$\underline{D}(C_k) = \sum_{i \in C_k} \sum_{j \in C_k} A_{ij} \tag{2.4}$$

which respectively count the number of links between nodes in $C_k$ to nodes not in $C$ and the number of links from nodes in $C_k$ to other nodes also in $C_k$. Note that $\underline{D}(C_k)$ is the twice the number of intra-connections in $C_k$, as each edge is counted twice in the undirected graph. In a similar manner we define the external degree for a *node* $v_i$ in community $C_k$:

$$\overline{d}(v_i, C_k) = \sum_{j \notin C_k} A_{ij}. \tag{2.5}$$

Likewise, the *internal degree of node* $v_i \in C_k$ is

$$\underline{d}(v_i, C_k) = \sum_{j \in C_k} A_{ij}. \tag{2.6}$$

Then

$$d(v_i) = \underline{d}(v_i, C_k) + \overline{d}(v_i, C_k) \tag{2.7}$$

is the degree of node $v_i$.

A node $v_i$ in a community $C_k$ is *strong* if

$$\underline{d}(v_i, C_k) > \overline{d}(v_i, C_k), \tag{2.8}$$

And it is a *weak* if

$$\underline{d}(v_i, C_k) < \overline{d}(v_i, C_k), \tag{2.9}$$

**Strong and Weak Communities**

A community $C_k$ is termed *strong* [Radicchi et al., 2004] if

$$\underline{d}(v_i, C_k) > \overline{d}(v_i, C_k), \quad \forall i \in C_k. \tag{2.10}$$

That is, all nodes in a strong community make more connections to other nodes in the community than they do to nodes in other communities. A community is *weak* if there are more internal connections between nodes in the community than there are connections to external nodes:

$$\sum_{i \in C_k} \underline{d}(v_i, C_k) > \sum_{i \in C_k} \overline{d}(v_i, C_k). \tag{2.11}$$

The summation is used here to compare all the internal degree of nodes within the community with all the external degree of these nodes. These nodes could be either weak or strong while Equation 2.10 ensures that each node in community $C_k$ should

be strong.

Clearly, a strong community is also a community in the weak sense, but a community is not necessarily either weak or strong.

**Degree distribution** is the distribution of node degrees in the network $p(d)$. In real networks, these degrees are high for some nodes and low for others, and this distribution often follows power law $p(d) \propto d^{\alpha}$ where $\alpha$ is a constant. This type of the network is called a scale-free network, in contrast to random networks where edges are put randomly between any pairs of nodes. Therefore, the degree distributions in random networks are homogeneous [Lancichinetti et al., 2008]. Many real-world networks are scale-free such as the Internet, World Wide Web and others [Broder et al., 2000]. Scale-free networks are inhomogeneous where many nodes have a few connections and a few nodes have large connections [Wang and Chen, 2003].

**Shortest path**, or geodesic distance, is the minimum number of edges in any path between two given nodes in a graph. Dijkstra's algorithm could be used to find the shortest paths in the graph [Cherkassky et al., 1996].

**Average path length** measures the average number of edges between all possible pairs of nodes in the network [Albert and Barabási, 2002]. Suppose, we have a graph that consists of $N$ nodes and $\delta(i, j)$ is the distance between $v_i$ and $v_j$. The average shortest path length is calculated as:

$$l = \frac{1}{N(N-1)} \sum_{i,j \in V} \delta(v_i, v_j) \tag{2.12}$$

**Betweenness Centrality** represents the number of times that a node is passed through on the shortest path between two other nodes [Freeman, 1977; Wasserman and Pattison, 1996; Brandes, 2001; Faust, 1997]:

$$BC(v) = \sum_{i \neq j \neq v \in V} \frac{\sigma_{i,j}(v)}{\sigma_{i,j}} \tag{2.13}$$

where $\sigma_{i,j}(v)$ represents the number of shortest paths between two nodes $i$ and $j$ that involved node $v$. $\sigma_{i,j}$ is the number of possible shortest paths between two nodes $i$ and $j$.

**Closeness Centrality** is defined as the reciprocal of the distance of a node to all other nodes in the network [Preparata et al., 2008]:

$$CC(v) = \frac{N-1}{\sum_{i \in V} \delta(v,i)}, \quad i \neq v \tag{2.14}$$

Thus a node with a high closeness centrality is in some senses close to the centre of the network.

## 2.2 Community Structure Evaluation Scores

**Community structure** is an important feature in complex networks where this network is divided into groups of nodes that have dense connections within the group and sparsely with the others. These groups represent a fundamental concept for analysing and understanding the complex networks, because analysing at a group level is easier than at a node level. These structures are most often found in real networks, for example in social networks, communities may represent common interests, or in biological networks, communities may refer to proteins that have similar functions [Scott, 2017; Lee and Lee, 2013; Newman, 2018]. The natural partition for a given network is based on connections between nodes. Each node is assigned to only one community, and nodes that have connections tend to be in the same community rather than nodes that are not connected. Finding community structure is a difficult issue as the number of communities and their size in real networks is unknown and the computational complexity of evaluating all possible partitions $O(2^N)$. Despite these difficulties, many algorithms have been proposed to unfold these communities as will be described later in sections 2.5, 2.6 and 2.7.

One of the valuable aspects that reflect an interesting investigation in network com-

munity detections is evaluation scores: that is, what type of score is suitable to evaluate community structures. We describe the most common score functions which have been used for quantifying how well a particular community structure fits a given network. The idea is that given a community structure, and the score function, we can evaluate whether this structure is fitted to a given network or not. Some of the scores are minimised while the others are maximised. However, all of them are formulated with the same intuition that there are dense connections within communities, while communities are sparsely connected with the other communities in the network.

Many scores have been introduced for evaluating network partitions. We focus on the following scores:

**Modularity**: Modularity measures the strength of partitioning a network into communities [Newman and Girvan, 2004]. Network partitions that have high values of modularity have dense connections within the community and sparse connections with the others. It is widely accepted as a score that has been used in optimisation methods for community detection in the networks. Modularity is defined as:

$$Q(\mathcal{C}) = \sum_{k=1}^{K} \cdot \left[ \frac{D(C_k)}{2L} - \left( \frac{D(C_k)}{2L} \right)^2 \right] \tag{2.15}$$

$Q(\mathcal{C}_k)$ may be shown to be the summed differences between the fraction of links within a community minus the expected fraction of links within the community if the graph were rearranged at random but preserving the degree distribution [Newman and Girvan, 2004]. The range values for modularity falls in the range of (-0.5, 1) where 1 point to accurate community structures [Brandes et al., 2008]. The modularity value is positive if the number of connections with the community is more than the number of expected from a random arrangement in which the degree distribution is preserved. It is negative when each node is in one community (or sometimes when the network is partitioned into very small communities) and 0 when all nodes are in one community.

Much existing literature uses $Q$ for evaluating results. However, Fortunato and Barthelemy [2007] have shown that it may fail to identify clusters if their size is smaller than a scale which depends on the size of the network and the interconnections between clusters. They concluded that the generated partition due to modularity optimisation has a resolution limitation and optimising $Q(\mathcal{C})$ may generate partitions which fail to identify small communities.

**Community Fitness (CF)**: This score is proposed by Lancichinetti et al. [2009]:

$$CF(\mathcal{C}) = \sum_{k=1}^{K} \frac{\underline{D}(C_k)}{(\underline{D}(C_k) + \overline{D}(C_k))^{\alpha}} \tag{2.16}$$

where $\alpha$ is a positive value which controls the size of communities. If $\alpha$ is large then the network will be divided into small communities, while if it is small large communities will predominate. Therefore, the external connections between communities will be minimised when the CF gets a high value (when $\alpha$ is small).

**Normalised Cut (NC)**: This score minimises edge weights between clusters relative to degrees of a cluster [Dhillon et al., 2004]. It aims to minimise the external degree of community with respect to the internal and external degree of this community:

$$NC(\mathcal{C}) = \sum_{k=1}^{K} \frac{\overline{D}(C_k)}{\underline{D}(C_k) + \overline{D}(C_k)}. \tag{2.17}$$

Network partitions that have a small $NC$ produce good communities, as these communities are well connected within themselves and sparsely with the other communities in the network.

**Kernel K-Mean (KKM)**: This objective is related to Kernel K-Means clustering because $KKM$ is a decreasing function and can generate a small number of communities [Angelini et al., 2007]. The KKM score is defined as:

$$KKM(\mathcal{C}) = 2(N - K) - \sum_{k=1}^{K} \frac{\underline{D}(C_k)}{\mid C_k \mid}. \tag{2.18}$$

The second term is maximised by maximising the average internal degree of the clusters; subtracting this term from the sufficient minuend $(2(N-K))$ to consider this objective as a minimisation objective and make the result of this score as a positive value. For example, if the partition $\mathcal{C}$ is divided into $N$ communities that means each node in one community then the result is 0 due to the values of the first and second terms are 0. Otherwise, the result is positive. This score is used by Gong et al. [2014] for community detection.

**Ratio Cut (RC)**: This score is an adaption of the Normalised Cut to solve the community detection problem. It is minimised when there are few edges between clusters relative to the size of the cluster [Gong et al., 2014; Dhillon et al., 2004]:

$$RC(\mathcal{C}) = \sum_{k=1}^{K} \frac{\overline{D}(C_k)}{\mid C_k \mid}. \tag{2.19}$$

By minimising $RC$, partition with sparse connections between communities are produced.

**CONductance (CON)**: This score measures the fraction of edges that connect to the nodes out of community [Yang and Leskovec, 2015]. It is defined as follows:

$$CON(\mathcal{C}) = \sum_{k=1}^{K} \frac{\underline{D}(C_k)}{\underline{D}(C_k) + \overline{D}(C_k)}. \tag{2.20}$$

**Community Score (CS)**: The community score is an attempt to increase the weight of the degree of the internal nodes within community [Pizzuti, 2008]. $CS$ is calculated by the summation of the local score for each cluster:

$$CS(\mathcal{C}) = \sum_{k=1}^{K} \frac{1}{\mid C_k \mid} \sum_{v \in C_k} \left( \frac{\underline{d}(v, C_k)}{\mid C_k \mid} \right)^r \times \underline{D}(C_k) \tag{2.21}$$

Where $r$ controls the size of communities, as an attempt to increase the weight of the degree of the internal nodes within the community.

**Internal Density (ID)**: The internal density measures the density of the internal

degree of the community [Yang and Leskovec, 2015].

$$ID(\mathcal{C}) = \sum_{k=1}^{K} 1 - \frac{\underline{D}(C_k)}{\mid C_k \mid (\mid C_k \mid -1)} \tag{2.22}$$

Thus maximising $ID(\mathcal{C})$ yields communities with strongly connected nodes within each community.

Although many objective functions have been proposed, there is little literature evaluating the quality of community detection scores. In 2015, Yang and Leskovec proposed an exciting method to evaluate quality scores based on the ground-truth partition [Yang and Leskovec, 2015]. However, this method has not presented how well correlated the optimising score on the generated partitions and evaluating measure between generated and correct partitions is. Hafez et al. investigated the performance of different objectives for community detection using single and multi-objective evolutionary algorithms [Hafez et al., 2014]. They used a single objective to optimise each of the existing objectives separately, and the authors used MOEA to optimise each pair of these objectives.

In general, all objectives have the same aim: either the objective is to increase the number of connections within the community (intra-connections) or decrease the number of connections with the rest (inter-connections). Later in chapter 3, we investigate the accuracy of these scores in assessing the difference between a partition and the true partition in terms of the Normalised Mutual Information [Danon et al., 2005].

## 2.3 Multi-Objective Optimisation Problems

In this thesis, we attempt to generate a set of network partitions with dense intra-connection and sparse inter-connection by optimising two conflicting objectives simultaneously. Therefore, we now describe the basic ideas of multi-objective optimisation. An Optimisation algorithm attempts to find the best solution among

many feasible solutions under the specific constraints. Some real-world problems require more than one objective function to be optimised simultaneously. This process is called Multi-Objective Optimisation (MOO) which produces a set of trade-off optimal solutions, called Pareto-optimal solutions [Coello et al., 2007].

Consider the following Multi-Objective Optimisation Problem (MOP) which seeks to simultaneously minimise $m$ objectives:

$$\text{Minimise} \quad F(\mathcal{C}) = (f_1(\mathcal{C}), f_2(\mathcal{C}), ...f_m(\mathcal{C})) \tag{2.23}$$

where $\mathcal{C} \in \Omega$ is a solution and $\Omega$ denotes the space of feasible solutions. If the objectives are competing, then at the optimum any improvement in one objective must diminish the performance on at least one other. This idea is made precise through the notion of dominance. Given two solutions $\mathcal{C}_1$ and $\mathcal{C}_2, \mathcal{C}_1$ is said to dominate $\mathcal{C}_2$ (denoted as $\mathcal{C}_1 \prec \mathcal{C}_2$) iff

$$\forall i \quad f_i(\mathcal{C}_1) \leqslant f_i(\mathcal{C}_2) \quad \wedge \exists i \quad f_i(\mathcal{C}_1) < f_i(\mathcal{C}_2). \tag{2.24}$$

If neither solution dominates the other they are said to be mutually non-dominating. A solution $\mathcal{C}^* \in \Omega$ is a Pareto-optimal solution to the minimisation problem (2.23) if it is not dominated by any other feasible solution. The set of all Pareto optimal solutions is named the *Pareto Set* (PS) and the image of the Pareto set under $F$ is known as the *Pareto front*. For these solutions, an improvement in one objective makes a degradation on at least another one. The goal of practical search algorithms is to produce a set of mutually non-dominating solutions that approximate the Pareto set. We use this method of optimisation to capture the community structures and evolution of communities in chapters 3 and 4.

In the last two decades, *Genetic Algorithms* have played a useful and vital role in optimising multiple objectives to solve real-world problems in different domains [Konak et al., 2006]. A Genetic Algorithm (GA) is an optimisation and search method which

is proposed by Holland [1975] to generate solutions based on biologically-inspired operators (selection, crossover and mutation). It evolves a population of chromosomes where each chromosome has a fitness value which is the result of computing the objective function to evaluate each chromosome separately. The fitness function has a vital role to develop the genetic algorithm, for more details about GA background see [Bäck et al., 1997; Goldberg and Holland, 1988; Boyd and Vandenberghe, 2004; Konak et al., 2006; Corne and Lones, 2018]. GA has been developed to optimise more than one objective. In this case, multi-objectives are optimised by an evolutionary algorithm to find a set of feasible solutions called Pareto optimal set [Coello et al., 2007; Zhou et al., 2011].

## 2.4 Multi-Objective Evolutionary Algorithms

In last few decades, evolutionary algorithms (EAs) have been successfully used for optimization problems involving more than one conflicting objective, as these algorithms are capable to produce a set of solutions in a single run. This set of solutions are an approximation to the Pareto-optimal set, as described above. In 1985, the first real multi-objective genetic was proposed by Schaffer [1985]. This approach involves generating subpopulations at each generation. The number of subpopulations are equal to the number of objectives. Each sub-population is responsible for searching one objective. Since 1985, different evolutionary algorithms have been proposed for multiobjective optimisation using evolutionary algorithms (MOEAs). For example, [Fonseca et al., 1993; Srinivas and Deb, 1994; Fonseca and Fleming, 1996; Zitzler, 1999; Zitzler and Künzli, 2004; Zhou et al., 2011]

Multiobjective evolutionary algorithms can be classified into three groups:

**Dominance-based algorithms**: These are the most popular multi-objective evolutionary algorithms that have been proposed by many researchers. In 1999, Zitzler and Thiele [1999] proposed the Strength Pareto Evolutionary Algorithm (SPEA) for

approximating the Pareto-optimal set for multi-objective optimization problems. It combined different features from the previous multi-objective EAs in one algorithm. For example, it stores a non-dominant evaluated set of solutions in an external population, assigns scalar fitness values to individuals based on the Pareto dominance concept and if the size of the Pareto set is larger than a predefined limit, then the Pareto set is reduced without destroying its characteristics by using clustering methods.

In 2000, Knowles and Corne proposed the Archived Evolution Strategy (PAES) that uses the dominance concept to evaluate solutions [Knowles and Corne, 2000]. It is used a single-parent single-offspring EA similar to a (1+1) evolution method. They used binary strings and bitwise mutations to create offsprings. This algorithm compares the offspring with respect to the parent. If the parent is dominated by the offspring, the offspring is the next parent while if the offspring is dominated by the parent, the offspring is rejected and it finds a new one. On the other hand, if neither dominates the other, both the parent and the offspring are compared with an archive of best solutions found so far.

The Nondominated Sorting Genetic Algorithm II (NSGA-II) [Deb et al., 2002] is a commonly used MOEA. It is an elitist algorithm in which the approximation to the Pareto set (the maximal set of non-dominated solutions) is kept from generation to generation. A crowding distance strategy is used to limit the size of the Pareto set and help improve the spread of solutions across the Pareto front. The crowding distance strategy is used to perform density estimation of solutions surrounding a specific solution in the population and create a Pareto rank for each individual. Nondomination individual rank and crowding distance are needed to create a population of individuals. In addition, there are other algorithms based on dominance concepts such as the Niched Pareto Genetic Algorithm (NPGA) [Abido, 2003] and Multi-objective Differential Evolution (MODE)[Varadarajan and Swarup, 2008].

**Indicator based algorithms** The main issue with multi-objective evolutionary al-

gorithms is the approximation of the Pareto optimal set. Indicator based approaches use a scalar indicator such as hypervolume and generational distance to measure the quality of the Pareto front [von Lücken et al., 2014]. These algorithms used the indicator to direct the search. Zitzler and Künzli [2004] proposed the first indicator-based an evolutionary algorithm (IBEA). In this algorithm, a pair of solutions are compared using a binary indicator and it does not need any diversity preservation mechanism. In 2005, Emmerich et al. [2005] proposed S-metric selection-EMOA (SMS-EMOA) which is based on the hypervolume measure to combine the concept of a selection operator and non-dominated sorting. It is similar to NSGA-II except in selection and there is a different ranking method used for the Pareto optimal solutions. This algorithm is designed to maximise hypervolume which is the size of dominated space [Hopfe, 2009]. However, the complexity of computing the hypervolume indicator in high dimensions is expensive. This algorithm showed good results for two or three objective problems. In order to deal with this problem Bader and Zitzler [2011] proposed the approximation of exact hypervolume values using a Monte Carlo algorithm and they presented this idea as Hypervolume Estimation Algorithm for Multi-objective Optimization (Hype).

**Decomposition based algorithms**: Another promising multi-objective evolutionary algorithm for optimising multi objectives by using the scalar functions are decomposition algorithms [Jaszkiewicz, 2004; Hughes, 2007; Li and Zhang, 2006; Zhang and Li, 2007]. Two difficulties associated with solving multi-objective problems need to be determined: 1- The number of solutions to approximate the Pareto front increases exponentially [Ishibuchi et al., 2008]. 2- The ability of search will deteriorate. The advantage of these algorithms is computational efficiency for calculating the scalarisation function. One of the most popular decomposition methods is MultiObjective Evolutionary Algorithm based on Decomposition (MOEA/D which has been developed by Zhang and Li [2007]. A lot of literature demonstrated that MOEA/D has a high ability of search on different test problems and it has a low computation of complexity [Chang et al., 2008; Ishibuchi et al., 2009, 2010; Konstantinidis et al.,

2010; Li and Zhang, 2009; Peng et al., 2009; Zhang et al., 2010]. The main idea behind MOEA/D is to decompose the multiobjective optimisation problem into a number of scalar optimisation subproblems rather than solving a MOP as a whole. There are three approaches which have been used by Zhang and Li [2007] for the decomposition process. The simplest one is the weighted sum aggregation method. This method works well when the Pareto Front (PFs) is concave, but that disadvantage is the nonconcave Pareto front can not be handled. The second one is the Boundary Intersection method which is used with nonconcave PFs. The last one and most popularly used is the Tchebycheff approach. This approach can be used with fronts that contain concave and convex regions. We have therefore used this method for the work presented here and the algorithm is described in more detail in Chapter 3.

In this thesis, we will use MOEA/D to optimise two conflicting objectives simultaneously to produce a set of candidate solutions in evolutionary optimisation method; this will be discussed in chapters 3 and 4.

## 2.5 Survey of Network Community Detection.

In this section, we survey the existing methods for community detection in a given network. These methods attempt to divide $G$ into small groups of nodes based on their relations. If the number of edges within these groups is large and between these groups are small, then a good partition could be generated.

In recent years, many efficient techniques have been proposed to unfold communities structure in the complex networks [Gong et al., 2014; Shi et al., 2012; Pizzuti, 2012; Wu and Pan, 2015; Zhao et al., 2018]. At the heart of all these methods lies the definition of a score function to evaluate the quality of a candidate partition. Many authors also propose methods to generate "good" candidate partitions.

### Hierarchical methods

Hierarchical methods are traditional methods that have been used to reveal the structure of communities. In real-world, many graph structures have been represented as dendrograms where each dendrogram displays a multilevel structure. Each level reflects the grouping of nodes with small groups inside large groups, which are in turn within larger groups, etc. There are two types of strategies for constructing hierarchical structures: agglomerative and divisive hierarchical algorithms. Agglomerative algorithms (bottom-up) start with every single node forming one cluster as the initial partition. After that, in each iteration, the most similar pair of clusters are merged and so on until all clusters are merged into one cluster [Jain et al., 1999]. Divisive algorithms (top-down) reverse the agglomerative algorithms: it considers all vertices as one big cluster initially. Recursively a division is implemented as each iteration moves down a level by removing edges. There are two advantages of the hierarchical methods:1) There is no need to specify in advance the number or size of the clusters in the network. 2) It can find a large number of network partitions. On the other hand, the disadvantage of this method that it can not correct a mistake made in early iterations. A similarity between clusters measure is required for each generation in both algorithms.

In 2002, Girvan and Newman used a divisive clustering method to detect community structure in biological networks [Girvan and Newman, 2002]. This process was performed by calculating the edge betweenness; the edge with the highest edge betweenness is removed, and betweenness recomputed. The computational complexity of this method is $O(N^3)$ due to the cost of computing betweenness for all edges which makes this algorithm impractical for large networks.

In 2004, Newman and Girvan improved this method by maximising the modularity [Newman and Girvan, 2004]. Divisive clustering was again used to iteratively remove edges with high edge betweenness to divide the network into communities, and the authors chose the partition which maximised the modularity. However, the time

complexity $O(L^2N)$ restricts this procedure from treating large networks. In the same year, Newman also proposed an agglomerative clustering algorithm to greedily maximise modularity without computing edge betweenness [Clauset et al., 2004]. While this algorithm has the advantage that the number of communities does not have to be pre-specified, the greedy nature of the search means that sub-optimal partitions may be located.

### Evolutionary Algorithms

Many real-world problems have been solved using multi-objective evolutionary algorithms (MOEAs). The network can be clustered into communities and this clustering process can be formulated as an optimisation problem. Recently, a group of scientists have been working on community detection using evolutionary algorithms. In 2008 Pizzuti used a Genetic Algorithm (GA) for community detection, potentially avoiding some of the problems associated with the greedy search which can become stuck in local minima [Pizzuti, 2008]. Pizzuti used a Single-Objective Evolutionary Algorithm (SOEA) to optimise the community score (Equation 2.21). However, the disadvantage of using single objective is that it may be biased on community partition which is obtained through optimisation process [Shi et al., 2014]. In this case, the network partition is generated with the fixed property. Recognising that many alternative definitions of community quality are possible, Pizzuti developed her work by using the multi-objective evolutionary optimisation to approximate the optimal trade-off between more than one measure of community quality[Pizzuti, 2012]. She proposed MOGA-Net which employed NSGA-II for community detection in networks. The author formulated community detection as a two-objective optimisation problem. The first objective is Community Score (CS) (Equation 2.21) and the second one is Community Fitness (CF) (Equation 2.16), proposed by Pizzuti [2008] and Lancichinetti et al. [2009] respectively. The $CS$ objective maximised by partitions with many connections within the same community (intra-connections), while the $CF$ objective is optimised by partitions with few inter-connections.

Solutions to the multi-objective problem are partitions which are globally non-dominated: that is, no other feasible partition has a wholly better $CF(\mathcal{C})$ and $CS(\mathcal{C})$. Thus this set of solutions – known as the Pareto set – represents the optimal trade-off between partitions optimising $CS$ and $CF$ and generally presents a variety of different possible community structures. The results showed that MOGA-Net reflects a good performance to produce accurate community structures compared with the state-of-art methods at that time. Therefore, many researchers were encouraged to develop MOEA for community detection as a set of near-optimal solutions (different community structures) will be generated rather than one solution that is generated using SOEA.

In the same year, Shi et al. also formulated community detection as a multi-objective minimisation problem [Shi et al., 2012]. They divided modularity (Equation 4.2) into two terms as these terms describe conflicting properties of the structures of the communities, measuring the degree of intra-connection and the degree of inter-connections. They, therefore, define two objectives to be minimised as follows. The first measures the intra-connections:

$$\text{Intra}(\mathcal{C}) = 1 - \sum_{k=1}^{K} \frac{\underline{D}(C_k)}{2L}, \tag{2.25}$$

The maximum value for $\frac{\underline{D}(C_k)}{2L}$ is 1 when all nodes in one community. The maximum value for $\frac{\underline{D}(C_k)}{2L}$ is 1 when all nodes in one community ($D$ is double of $L$). The authors subtracted the first term of modularity from 1 to formulate this problem as a minimisation optimization problem and 1 is sufficient to make the result of this score as a positive value or zero. The second objective measures the degree of inter-connections:

$$\text{Inter}(\mathcal{C}) = \sum_{k=1}^{K} \left[ \frac{D(C_k)}{2L} \right]^2 \tag{2.26}$$

Shi et al. showed that the simultaneous optimisation of these two objectives can yield a wide range of possible community structures, placing more or less weight on intra and inter-community connections. Since the modularity is the sum of these

two objectives, it is clear that the partition that maximises the modularity must be a member of the Pareto set.

In 2014, Gong et al. introduced a Discrete Particle Swarm Optimization algorithm to unfold the structure of communities in the networks [Gong et al., 2014]. In this method minimisation of the first objective is Kernel K-Means (Equation 2.18) which maximises the average internal degree of the clusters: The second objective is the Ratio Cut (Equation 2.19) which is minimised when there are few edges between clusters relative to the size of the cluster. Like the other multi-objective optimisation algorithms, simultaneous optimisation of the Ratio Cut and Kernel K-Means objective results in a set of solutions trading off partitions with a high degree of intra-community connectedness with partitions possessing few inter-connections.

Recently, Cheng et al. developed a multi-objective evolutionary algorithm, termed LMOEA to solve the community detection problem [Cheng et al., 2018]. Two conflicting objectives are optimised in this algorithm: Negative Ratio Association (NRA) and Ratio Cut (RC) (Equation 2.19), respectively. The NRA is defined as:

$$NRA(\mathcal{C}) = -1 \times \sum_{k=1}^{K} \frac{\underline{D}(C_k)}{\mid C_k \mid}. \tag{2.27}$$

Mininmising the NRA promotes partitions with communities that have a high proportion of internal connections. This objective is conflict to the $RC$ which minimises the connections between communities.

Empirical results indicated that this algorithm can detect the community structures with high quality.

In chapter 3, we will introduce a new two conflicting objectives are optimised simultaneously using an MOEA/D to unfold more accurate community structures and we will propose a new heuristic strategy as a mutation operator to speed up the converge our algorithm.

### Stochastic Block Model

We review the Stochastic Block Models (SBMs; [Holland et al., 1983]) which have been used in complex network analysis. SBM is classified as a random graph model which represents a generative model for communities in networks to fit the observed adjacency matrix by the maximization of a likelihood (generative models). In SBM, each node is assigned to one block or community. Links between paired nodes are generated according to probabilities which depend on cluster memberships of the connecting nodes. This method of clustering has been used in literature for community detection in networks [Amini et al., 2013; Bickel and Chen, 2009; Karrer and Newman, 2011].

Hofman and Wiggins proposed a general Bayesian approach infer community assignments where each observed link is modelled with a mixture of Bernoulli distributions and a community label for each node is assigned with a prior probability [Hofman and Wiggins, 2008]. Newman and Leicht proposed a mixture model with the expectation-maximization algorithm to model the community structure of networks [Newman and Leicht, 2007]. The authors classified nodes into groups based on the observed connections between them. In general, these studies model the distribution of nodes and determine the structures of communities. However, most SBM methods do not consider the distribution of the degree of nodes as these methods generate edges randomly between nodes. Karrer and Newman proposed degree-corrected SBM [Karrer and Newman, 2011]. They incorporated degree heterogeneity into block models. In this case, expected degrees close to the observed degrees.

SBM is also used for detecting communities and their evolution in dynamic networks. In 2008, Lin et al. proposed FacetNet for analysing communities and their evolution in dynamic networks [Lin et al., 2009]. This method is the first probabilistic generative model to address the problem of evolutionary clustering based on a probabilistic perspective. These methods require high memory.

### Spectral clustering

Spectral clustering uses the eigenvectors of matrices to partition a network into clusters. The initial set of objects are transformed into a set of points, elements of eigenvectors are coordinated for these points. The traditional clustering methods could be used to cluster these points (for example, k-means). The first spectral clustering was by Donath and Hoffman [1973]. The most popular spectral approaches are unnormalized spectral clustering which has been proposed by Shi and Malik [2000] and normalized spectral clustering methods [Ng et al., 2002]. This method fails to cluster the datasets that have different structures at density and size scales [Nadler and Galun, 2007]. For an extensive review of spectral clustering see [Von Luxburg, 2007]

### Algorithms based on Modularity

Modularity measure (Equation 2.2) has been proposed by Newman and Girvan [2004] is the most popular and best known quality measure which has been used by many scholars in community detection algorithms. We will classify clustering methods based on modularity as follows:

- Greedy algorithm: The greedy algorithm is the first algorithm that has been used to maximise modularity by Newman and Girvan [2004]. It is an agglomerative algorithm as we discuss on page 25. Later on, work improved the speed of the Newman and Girvan algorithm by using the max-heaps data structures [Clauset et al., 2004]. Although this algorithm is fast, it is biased to large communities. Danon et al. [2006] suggested a better modularity optima (in terms of community size) compared with the previous one by normalising the variation in modularity. This normalisation was accomplished by the merging of pairs of communities by the fraction of edges incident to one of the pair communities. In 2008, Blondel et al. [2008] proposed a different greedy algorithm (it is known as Louvain algorithm) to find communities in weighted networks.

This method starts by considering each node as a community and merging these communities based on the maximising of modularity. This process is repeated on the set of nodes until a maximum of modularity is reached. This method is low in time complexity. However, it depends on the order in which nodes are visited. As a result, the greedy optimisation tends to be inaccurate.

- Simulated annealing: Simulated annealing [Kirkpatrick et al., 1983] is a probabilistic method for a global optimisation to find an approximate global optimum in the search space. In 2004, simulated annealing was used by Guimera et al. [2004] for the first time to find the best network partition by maximising modularity. Its base implementation, Guimera and Amaral [2005] is where a single node is randomly selected and shifted from one community to another. A global movement is applied by splitting and merging communities using computational temperature to avoid trapping in local minima [Massen and Doye, 2005]. These methods accurately detected community structures, but were very expensive in terms of time complexity.

- Other optimization methods: A framework of mathematical programming was developed to maximise modularity by Agarwal and Kempe [2008] as the optimization of modularity can be formulated as a linear program. Mathematical programming approaches are promising but the limitation of this method is high computational complexity. White and Smyth [2005] used spectral clustering approach for modularity optimisation. Brandes et al. [2008] maximised modularity by an integer programming formulation to facilitate optimization without enumeration of all clusters.

### Random walk

Random walk [Hughes, 1995] has been used by several algorithms for community detection. The idea is to find the similarity between nodes based on a random walk. Random walk overcomes the limitation of finding similarity (distance) between nodes

based on the shortest path. The shortest path based distance is six-degree separation which works well with small world networks [Newman, 2008] but it does not work with large social networks. In 2003, Zhou used random walks to find a distance between two nodes [Zhou, 2003]. The distance is the average number of links that a random walker visits from node $i$ to node $j$. The nodes that have small distance are more likely to be in the same community. If the community is strong then the random walk consumes more time within the community since this community has dense connections.

In 2006, a different distance measure was introduced by Latapy and Pons [Pons and Latapy, 2006]. The authors proposed an algorithm which is called the walk trap community detection algorithm. The authors used a random walk to find similarity between nodes in the graph based on diffusion distance (a random walk). The distance which is defined as the probability of random walker moves from one node to another (one of its neighbour in one step) with a constant number of steps. The transition probability from node $i$ to node $j$ at each step is $P_{ij} = \frac{A_{ij}}{d(v_i)}$, $P = D^{-1}A$, where $D$ is a diagonal matrix with $D_{ii} = d(v_i)$. $P_{ij}^t$ is the probability of random walk from node $i$ to node $j$ in $t$ steps. The length of $t$ ($t$ is the number of time steps) should be sufficient to find important information about the network. The distance between two nodes is calculated by the following equation:

$$r_{ij}(t) = \sqrt{\sum_{u=1}^{N} \frac{(P_{iu}^t - P_{ju}^t)^2}{d(v_u)}} \tag{2.28}$$

This similarity measure is used in an agglomerative method where communities are merged based on the short random walk as existing in one community is better than leaving it. The modularity is used to select the best partition from the resulting structure. Although this approach works well to capture the information on the community structure, it needs more memory space.

## 2.6 Evaluation Measures

To evaluate the quality of generated partitions against ground-truth partitions an evaluation measure is needed. There are several measures to evaluate how well the network partition matches the true partition. In this section, we present the commonly used measures for comparing the detected network partitions with ground-truth partition.

**Purity** is a simple external criteria for evaluating cluster quality and the first measure that has been used in context of community detection [Zhao and Karypis, 2001; Schütze et al., 2008]. Let $P^*$ denotes the ground-truth partition whose communities are $\{P_j^*\}$ and $\mathcal{C}$ is the partition that found by community detection algorithm, then the $Purity$ measure can be calculated by the following equation:

$$Purity(\mathcal{C}) = \sum_{k=1}^{K} \frac{|C_k|}{N} \max_j Precision(C_k, P_j^*) \tag{2.29}$$

The precision is defined as:

$$Precision(C_k, P_j^*) = \frac{|C_k \cap P_j^*|}{|C_k|} \tag{2.30}$$

The purity measure is not symmetric which leads researchers to take the harmonic mean of $Purity(\mathcal{C}, P^*)$ and $Purity(P^*, \mathcal{C})$. Higher purity corresponds to a better match between the partitions. However, the maximum Purity value can be achieved if each node form one community as the Purity measures biases to the partition that has small cluster size. Purity does not reward grouping elements from the same class together. Each individual cluster has a Purity and any change inside other clusters do not change the Purity of that individual [Amigó et al., 2009]. Despite these disadvantages, it is still considered as an important measure as it provides an important information about clustering evaluation.

**Inverse Purity ($IP$)** is proposed to use the cluster with maximum recall for each

cluster in the true partition. It rewards grouping elements together. This measure is defined as:

$$IP(\mathcal{C}) = \sum_{k=1}^{K} \frac{|P_k^*|}{N} \max_j Precision(P_k^*, C_j) \tag{2.31}$$

**F measure (harmonic mean)** is combined by both Purity and Inverse Purity [Van Rijsbergen, 1979]. It is defined by the following equation:

$$F = \sum_{k=1}^{K} \frac{|P_k^*|}{N} \max_j F(P_k^*, C_j) \tag{2.32}$$

where

$$F(P_k^*, C_j) = \frac{2 \times Recall(P_k^*, C_j) \times Precision(P_k^*, C_j)}{Recall(P_k^*, C_j) + Precision(P_k^*, C_j)} \tag{2.33}$$

$$Recall(P^*, \mathcal{C}) = Precision(\mathcal{C}, P^*) \tag{2.34}$$

Exactly matching partitions have an F-score of 1, while the minimum value is 0 for partitions that do not intersect. The F score prefers coarse clustering in contrast to purity which prefers small size clusters. However, this measure has the same problem as purity where the F score is calculated for each individual clusters, so the individual score will not be affected by any change in other clusters [Amigó et al., 2009].

**Jaccard Index** is the similarity measure to compare two partitions. It is defined by dividing the size of intersection with respect to the size of the union [Jaccard, 1908],

$$J(\mathcal{C}, P^*) = \frac{\mathcal{C} \cap P^*}{\mathcal{C} \cup P^*} \tag{2.35}$$

**Normalised Mutual Information** [Danon et al., 2005] is the most widely used similarity measure to assess the accuracy of community detection algorithms. The $NMI$ has been proven to be reliable [Lancichinetti and Fortunato, 2009]. The $NMI$ value increases gradually when the two partitions become more similar and vice versa. In addition, $NMI$ is symmetric and unbiased in terms of the cluster distribution.

Let $P$ and $\mathcal{C}$ be two partitions of a network with $K_P$ and $K_\mathcal{C}$ communities respectively. Also, let $Z$ be the confusion matrix whose elements $Z_{ij}$ are defined as the number of nodes in community $i$ of partition $P$ that are also in community $j$ of partition $\mathcal{C}$. If $Z_i^P = \sum_j^{K_\mathcal{C}} Z_{ij}$ is the number of nodes in community $i$ of partition $P$ and similarly for $Z_j^B$, then the $NMI$ is defined as follows:

$$NMI(P,\mathcal{C}) = \frac{-2\sum_{i=1}^{K_P}\sum_{j=1}^{K_\mathcal{C}} Z_{ij}\log(Z_{ij}N/Z_i^P Z_j^C)}{\sum_{i=1}^{K_P} Z_i^P \log(Z_i^P/N) + \sum_{j=1}^{K_\mathcal{C}} Z_j^C \log(Z_j^C/N)} \tag{2.36}$$

The $NMI$ is non-negative and equal to zero if and only if the joint distribution $Z_{ij}/N$ can be written as a product of the distributions $Z_i^P/N$ and $Z_j^\mathcal{C}/N$, that is if knowledge of the partition $P$ provides no information about membership of partition $\mathcal{C}$. The $NMI(\mathcal{P},\mathcal{C}) = 1$ when $P$ and $\mathcal{C}$ are identical up to relabelings of the communities.

Therefore, we choose $NMI$ as a measure of the similarity between the known correct partition and a detected one as it can overcome the problem of comparing different community structures.

If the ground-truth partition for the network is unknown then the modularity is often used as the internal measure to assess the network partitions. However, it has a resolution limitation: it does not detect small communities well and tends to be skewed by the size of the whole network. To be more clear when the size of communities are small then each term of the modularity (see Equation 4.2) will be small thus the modularity value will be small. In this case, if the true partition is small communities, then the modularity never gets a large value [Fortunato and Barthelemy, 2007; Pizzuti, 2012; Lancichinetti and Fortunato, 2012; Miyauchi and Kawase, 2016]. Later in chapter 3 and 4, we use $Q$ and $NMI$ as evaluation measures for evaluating the goodness of the network partitions obtained by existing and proposed algorithms.

## 2.7 Community Detection in Dynamic Networks

In this thesis, we are interested in analysing the evolving communities over successive time steps. The networks are ubiquitous in many fields of science and society, ranging from computer science and mathematics to the biological and social fields. These networks have evolved rapidly over time such as Facebook, Twitter and LinkedIn [Hopcroft et al., 2004; Nguyen et al., 2014]. Dynamic networks can be represented as a sequence of snapshots at different time steps. Understanding different structures for communities in these networks (i.e. change over time) provides an opportunity to understand the configuration of the networks by analysing these networks. However, analysing and understanding these structures is extremely challenging due to the difficulty in tracking and detecting communities that change over time. Therefore, recently scholars put their attention towards studying temporal networks where nodes leave or join communities, a new community could appear or delete, a new edge connects existing nodes and edge removal over time. Figure 2.2 shows samples of the behaviour of dynamic communities evolve, for instance how the structure of communities can change (merge and expand) at successive times.

Although several algorithms have been proposed to detect communities and their evolutions in dynamic networks [Lin et al., 2009; Folino and Pizzuti, 2010, 2014; Xu et al., 2014; Ma et al., 2014], this problem is still open due to prompt changes in the structure of communities. Some of the proposed techniques detect communities in each snapshot independently and then track community evolving at different time steps [Leskovec et al., 2005; Kumar et al., 2005]. Despite the fact that these methods can track the evolution of communities, the weakness of these methods is analysing communities separate from their evolution produces community structures which tend to have a high difference [Lin et al., 2009]. As a result, these two steps produce undesirable community structure and evolution.

On the other hand, Chakrabarti et al. proposed the first evolutionary clustering

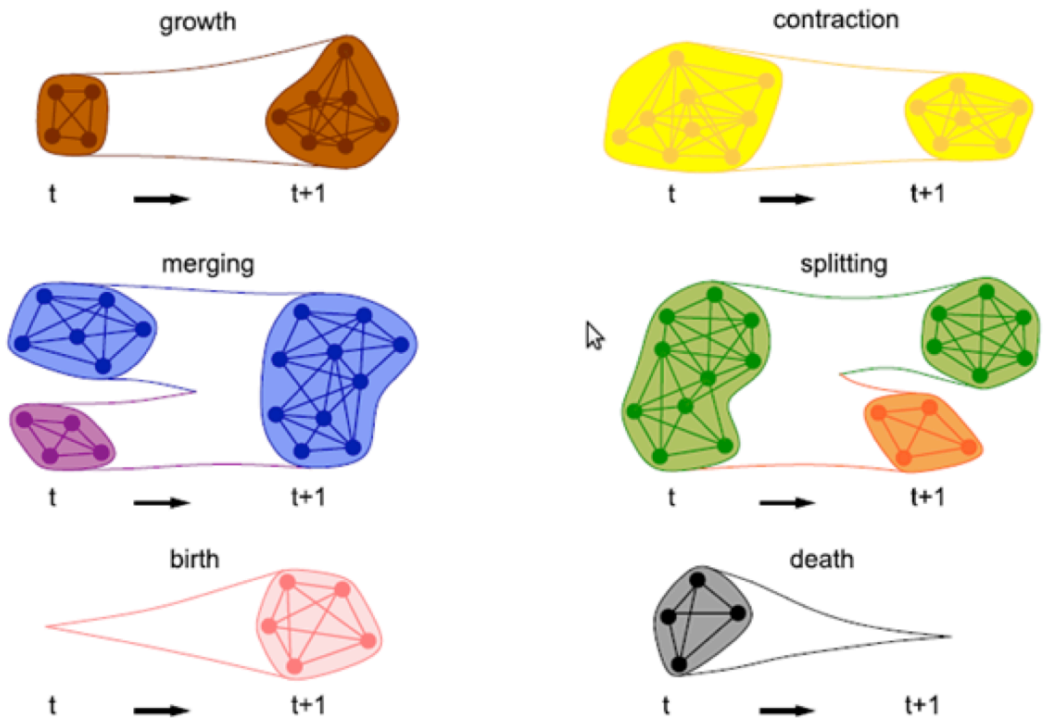


Figure 2.2 Possible structures for communities in dynamic networks [Palla et al., 2007].

framework to address the evolution of communities where the community at time $t$ is based on the community at time $t-1$ [Chakrabarti et al., 2006]. The meaning of evolutionary here is temporal evolution. Their formulation confers a temporal smoothness on the solution embodying the idea that dramatic changes in community structure from one time step to the next are undesirable. Their formulation is based minimising the weighted sum of two measures: two measures: Snapshot Cost ($SC$) and Temporal Cost ($TC$). Snapshot Cost measures how well a network is partitioned into communities. Temporal Cost measures the distance or dissimilarity between clusters at the current time and the previous one. The overall cost to be minimised is:

$$Cost = \alpha.SC + (1-\alpha).TC \tag{2.37}$$

Here $\alpha$ is a variable to control the preference of each sub-cost. This objective in Equation 2.37 became a source for many works of literature [Tang et al., 2008; Lin et al., 2009; Folino and Pizzuti, 2010, 2014] by performing a trade-off between snapshot cost and historical cost.

In 2008, Lin et al. proposed FacetNet for analysing communities and their evolution in dynamic networks [Lin et al., 2009]. The framework employs two models to capture the evolution of communities: the stochastic block model for generating communities and Dirichlet distribution to capture the evolution of communities. The snapshot cost is defined by using the KL-divergence to measure how to fit the approximate community structure that is computed by using a mixture model for the observed data. At each iteration, the value of the approximate structure is updated to decrease the cost function. This method converges to an optimal solution by the monotonic decrease of the cost function. However, the number of communities should be fixed over time.

Kim and Han proposed an efficient particle-and-density based evolutionary clustering method to address the problem of a variable number of communities over time [Kim and Han, 2009]. They introduced two concepts: nano-communities and l-clique-by-clique (l-KK). Nano-communities are a set of particles to model the dynamic network (which captures the evolution of communities over time) and l-clique-by-clique (l-KK) is a densely connected subset of particles (nano-communities) which form a community. Two nodes are connected if they are in different parties. Temporal smoothing is achieved using a cost embedding technique. The clustering method based density to partition the network. The algorithms that have been proposed by Lin et al. and Kim and Han, need specification of a parameter to control the preference to the snapshot quality or temporal quality.

In 2010, Folino and Pizzuti proposed a dynamic optimisation model using a multi-objective evolutionary algorithm [Folino and Pizzuti, 2010]. The authors used *Community Score* as the first objective that maximises the quality of community structure at the current time step while the second one $NMI$ that minimises the difference between the structures of communities over consecutive time steps as the dramatic shift between successive time steps is undesirable. They proposed the algorithm named DYNMOGA [Pizzuti, 2012] which employed NSGA-II [Deb et al., 2002] for this study. At each time step, a set of trade-off solutions between these two objec-

tives ($CS$ and $NMI$) are generated. They used modularity to select one solution among these set of solutions. In the next time step $NMI$ is calculated between the solution that has the highest modularity in the previous time step and the solutions at the current time step. This the first study to use MOEA to analyse the evolution of communities over time. Their results outperform the previous study such as [Lin et al., 2009] and [Kim and Han, 2009]. After that, in 2014, the same authors used different scores as the first objective such as modularity, Community Score, CONductance and Normalised Cut [Folino and Pizzuti, 2014]. The results showed that their algorithm has a good performance for detecting the dynamic communities specifically when $Q$ or $CS$ is used as the first objective

In the same year, Ma et al. employed a multi-objective evolutionary algorithm based on decomposition (MOEA/D) [Zhang and Li, 2007] to detect dynamic communities over time [Ma et al., 2014]. The authors also used $Q$ as the first objective to measure the quality of the structure of communities and $NMI$ as the second objective to measure the temporal cost. $NMI$ assesses the similarity between the best solution in the previous time step and the current community structures. Modularity density ($QD$) has been used to choose the best trade-off solution from the nondominated solutions at each time step.

$$QD(\mathcal{C}) = \sum_{k=1}^{K} \frac{\underline{D}(C_k) - \overline{D}(C_k)}{\mid C_k \mid} \tag{2.38}$$

$QD$ [Li et al., 2008] measures the ratio of the difference between internal and external degree corresponding to the size of the community. The partition maximising QD is chosen as the best partition at each timestep. This algorithm has a good contribution to capture community evolution. However, depending on just modularity for snapshot quality is not enough as we mentioned earlier that modularity has the resolution limitation problem.

In this section, we have reviewed the algorithms that have been used by the state-of-the-art to detect dynamic communities. We extend the work of Chakrabarti et

al. for evolutionary clustering to formulate this problem in a Hidden Markov Model (HMM) to capture the evolving of communities over time in chapter 4.

## 2.8 Summary

In this chapter, we have reviewed some of the basic concepts that are related to the objectives for evaluating and revealing the structure of communities in static and dynamic networks. In chapter 3, we propose two new conflicting objectives to discover community structures in synthetic and real-world networks.

As we can see, all related works are designed based on the relationship between communities except community score which it is design based on the relations between communities and between nodes as well, see Equation 2.21. According to our investigation, focusing on the relationship between nodes can provide valuable features about each node within the community. For example, the strong community is achieved by ensuring that each node should be strong rather than the summation of connections for all nodes within the community.

On the other hand, in the case of dynamic networks, all the existing methods for analysing community evolution have similar aspects which are snapshot quality and temporal cost. The authors used only one objective to evaluate snapshot quality such as modularity, Community Score, CONuctance, Normalised Cut, etc. [Folino and Pizzuti, 2010, 2014; Ma et al., 2014; Zhou et al., 2015] while community detection issue has been proved as multi-objective optimisation due to networks have multiple structural properties [Shi et al., 2012; Pizzuti, 2012; Gong et al., 2014; Wu and Pan, 2015].

In chapter 4 we, therefore, formulate the evolution of communities as a Hidden Markov Model in which the hidden states are found using a multi-objective algorithm, thus allowing a wide range of partitions to be considered and the Viterbi algorithm is used to find the most likely sequence of partitions over time.

# Chapter 3

# Community Detection in Static Networks

Detecting accurate community structures is important to understand the behaviour of the networks (see [Fortunato and Lancichinetti, 2009] for a review), i.e., a group of nodes that have dense connections within a community than the rest communities. Many algorithms have been proposed in the last two decades for community detection [Gong et al., 2014; Shi et al., 2012; Pizzuti, 2012; Wu and Pan, 2015; Zhao et al., 2018]. However, all these algorithms attempted to minimise all connections between communities without taking into account that there may be small connections between communities in the natural network partition. In addition, the proposed objective functions in the existing literature are designed based on community information for example, the number of connections inside community, the number of connections between communities, etc. rather than node information for example, the number of internal connections for each node within the community, the number of external connections for each node among different communities. If we ensure that as much as nodes in each community is, for example, strong then the detected partition will be tend to consist of strong communities. We note, however, that the *Community Score* considers the information at both community and node

level and therefore information at the node level has an effect on the evaluation of community structures. That motivates us to propose new objectives based on node relations within communities and between communities.

In this chapter, we formulate community detection as a multi-objective optimisation problem. A Multi-Objective Evolutionary Algorithm, named Multi-Objective Evolutionary Algorithm Based Community Detection in Networks (MOEA-CD) is used to optimise two new contradictory objectives simultaneously. This algorithm attempts to detect the structure of communities in static networks by employing the MOEA/D evolutionary algorithm [Zhang and Li, 2007], which has proved to be successful in solving Multi-objective Optimising Problems (MOPs) [Zhang and Li, 2007; Konstantinidis and Yang, 2011]. These references show that the MOEA/D algorithm outperforms or performs similarly to the most popular NSGA-II which has been proposed by Deb et al. [2002]. In general, Evolutionary Algorithms have demonstrated the possibility to reach global optima, and they do not need any prior knowledge which is very difficult to specify for real networks. Although the existing evolutionary algorithms for revealing community structure are effective, they need improvement to speed up convergence to the optimal solution. Also, there are a few studies to tackle this issue by combining local search technique with an evolutionary algorithm for community detection [Gong et al., 2014; Wu and Pan, 2015]. This issue motivates us to propose a new local heuristic search called the Neighbourhood Node Centrality ($NNC$) strategy to speed up the convergence of an EA to the optimal solution.

In addition, we propose a perturbation strategy that is different from perturbation strategies which have been proposed by Yang and Leskovec [2015] to evaluate the existing and new objectives by determining either the objective is strong or weak. The main contributions of this chapter are threefold:

1. A community structure score function evaluation technique is proposed based on a random migration strategy. This strategy is implemented by migrating

random nodes from original communities to random communities. The aim of this method to validate the quality of the existing and new scores.

2. A new multi-objective optimisation method is proposed to detect the structure of communities in real and synthetic networks. This model includes two new contradictory objectives to capture the intuition of community detection in the complex network system.

3. A new local heuristic search approach is suggested which is combined with our model to produce effective results.

As a consequence, we formulate the main milestones for our algorithm and provide an opportunity to produce a more accurate model to unfold the structure of communities against three current state-of-the-art models. This formulation will be presented in section 3.1 by introducing the formulation of our two objective functions.

In section 3.2 we describe our technique to assess the objective functions based on the ground-truth partition. Section 3.3 introduces our formulation for the community detection problem. Section 3.4 presents the proposed algorithm for network clustering. In section 3.5 we evaluate our model against three existing models with and without a local heuristic search on synthetic and real-world networks. Finally, the conclusion is presented in section 3.6.

## 3.1 Objective Function Formulation

Section 3.2 evaluates a range of objective functions. In order to include our objectives, we formulate them here. We attempt to simultaneously minimise two objective functions, one quantifying the density of internal connections within communities and the other quantifying the sparsity of connections between communities.

Let $\mathcal{C}$ be a network partition that is divided into $K$ communities $\{C_i\}_{i=1}^{K}$. Then an

objective quantifying the average proportion of internal neighbours in a relative to the degree of the node is:

$$f_{Intra}(\mathcal{C}) = 2(N-K) - \sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{v \in C_k} \frac{\underline{d}(v, C_k)^2}{d(v)} \tag{3.1}$$

where $N$ is the number of nodes, $d(v)$ is the degree of node $v$ and the internal degree of $v \in C$ $\underline{d}(v, C)$ is the number of edges from $v$ to other nodes in $C$ (Equation 2.6); We refer to this objective as the *Intra-Score*. The second term is maximised by increasing the average number of internal neighbours; subtraction of this term from its maximum value $2(N-K)$. The minuend is used to consider this objective as minimisation objective and it is sufficient to make the value of this objective as positive value. The range of $\frac{\underline{d}(v,C_k)^2}{d(v)}$ is between 0 and $N-1$. It is 0 when the partition $\mathcal{C}$ is divided into $N$ communities and that mean each node in one community ($N = K$). Therefore to make the minuend is 0 we need to put $N-K$. It is $N-1$ when all the network is considered as one community. In this case, $N-K$ is equal to $N-1$ and we used 2 in the term $(2(N-K))$ to produce a positive value.

The second objective function quantifies the average maximum number of links between communities. Let $I(v, C_j)$ be the ratio of the maximum number of edges between node $v \in C_j$ and any other community and the internal degree of $v$:

$$I(v, C_j) = \frac{\max_{C_i \neq C_j} \sum_{w \in C_i} A_{vw}}{\max(\sum_{w \in C_j} A_{vw}, 1)} \tag{3.2}$$

Then the *Inter-Score* is defined as

$$f_{Inter}(\mathcal{C}) = \sum_{j=1}^{K} \frac{1}{|C_j|} \sum_{v \in C_j} I(v, C_j). \tag{3.3}$$

Clearly $f_{Inter}(\mathcal{C})$ is minimised by partitions comprising communities which make few connections to other communities. Simultaneous minimisation of $f_{Intra}(\mathcal{C})$ and $f_{Inter}(\mathcal{C})$ is not generally possible, but the set of partitions that trade-off one against the other contains good approximations to the true partition when it is known.

## 3.2 Empirical Evaluation of Objective Fidelity

Although many objective functions to detect and quantify community structure in networks have been proposed in the literature, it is unclear how well these objectives represent the Normalised Mutual Information ($NMI$) between a candidate partition and the correct partition. As we discussed in chapter 2 that the $NMI$ has been proven to be reliable [Lancichinetti and Fortunato, 2009] and it's value increase gradually when the generated and ground-truth partitions become more similar and vice versa. However the ground truth partitions have not always strong communities but in general, the structures of ground truth partitions are resemble the natural partitions (more connections within community and less between communities).

To ascertain which objectives are effective for identifying community structure, we assess a range of objectives by generating partitions, $P$, which are perturbations of a given ground-truth partition, $P^*$, and compare the objective $f(P)$ with $NMI(P, P^*)$. We aim to find objectives $f(P)$ which are well correlated with $NMI(P, P^*)$ so that an optimisation algorithm may use $f(P)$ as a proxy for $NMI(P, P^*)$.

Without loss of generality, suppose that $f(P)$ is to be minimised, then it is desirable that $f(P) < f(P')$ if and only if $NMI(P, P^*) > NMI(P', P^*)$. In particular, we desire that there are no partitions for which $f(P) < f(P^*)$; we call such partitions *misleading*. One measure of the quality of an objective function is the fraction, $\mu_f$, of partitions in a sample for which $f$ is misleading.

One way of generating random partitions would be to assign each node in the graph to one of a fixed number of communities at random (we suppose the number of communities does not change). However, this procedure tends to generate partitions which are far from the true partition $P^*$, and we are especially interested in partitions close to $P^*$ (that is, with $NMI \approx 1$) because minimising a score function to find the "best" partition must distinguish between partitions close to $P^*$. We, therefore, generate partitions by reassigning the community of randomly cho-

**Algorithm 3.1** Method for objective evaluation based on ground truth partition. At each iteration $m$-node in the true partition, leave the original communities and migrate to random communities.

Inputs
1 : $P^*$ : Ground-truth partition
2 : $N$ : Number of nodes in the network
Steps
1 : $P^* = \{C_1, C_2, \ldots, C_K\}$ ▷ $K$ Maximum number of communities in $P^*$
2 : **for** $i = 1$ **to** $N$ **do**
3: **for** $j = i$ **to** $N - i + 1$ **do**
4 : **for** $m = 1$ **to** $i$ **do**
5 : $v \leftarrow rand(v_1, v_2, ..., v_N)$ ▷ $v$ is a random node that is selected from $P^*$.
6 : $C_v \leftarrow Community(v)$ ▷ $C_v$ is the community of the node $v$
7 : $NewC_v \leftarrow rand\{C_1, C_2, \ldots, C_K\}, \quad C_v \neq NewC_v$ ▷ $NewC_v$ is a random new community for the node $v$
8 : $Community(v) \leftarrow NewC_v$
9: **end for**
10: A new $m$ random partitions ($P^m$) are generated
11: **end for**
12: **end for**

sen nodes of $P^*$. Clearly, reassigning only a few nodes will yield partitions close to $P^*$, while reassignment of many nodes produces partitions distant from $P^*$, with small $NMI(P, P^*)$. The total number of possible partitions that could be generated is very large. We, therefore, adopt the following scheme to generate an ensemble of random partitions. $N$ partitions $P_i^{(1)}$ are generated by randomly selecting each node in $P^*$ and are assigned to a random community. Sets of partitions $P^{(m)}$ with $m$ randomly selected nodes and assigned to random communities, $m = 1, 2, ..., N$ as illustrated in Algorithm 3.1. The total generated partitions is $N(N + 1)/2$, with a greater number of partitions close to $P^*$ and smaller numbers more distant.

We evaluate the performance of new and existing score functions using the random sampling technique described above on six real-world networks: These networks are the Zachary karate club network [Zachary, 1977], the Bottlenose Dolphin network [Lusseau, 2003], the American football network [Girvan and Newman, 2002] and the Kreb's American politics network [Newman, 2006]. Figure 3.1 and 3.2 show plots of $f(P)$ versus $NMI(P, P^*)$ for the random sample of partitions and some different objective functions $f(P)$ (Modularity (Q), Community Score (CS), CONductance (CON), Normalised Cut (NC)[Folino and Pizzuti, 2010, 2014; Ma et al., 2014; Zhou

Table 3.1 The average of the number of misleading partitions over twenty runs for each objective. These objectives are tested on six real-world networks.

| **Objectives** | **Karate** | **Dolphin** | **Football2000** | **Football2001** | **Kreb books** |
|---|---|---|---|---|---|
| Q | 1 | 2 | 1.3 | 0.4 | 0.8 |
| CS | 0 | 0 | 0.3 | 0.1 | 0.8 |
| CF | 2.5 | 0 | 1.3 | 0.4 | 1.2 |
| KKM | 0 | 0.3 | 0.3 | 0.2 | 1.3 |
| Intra | 0 | 0 | 1.3 | 0.4 | 0.8 |
| Intra-Score | 0 | 0 | 0.3 | 0 | 0.7 |
| RC | 3.6 | 13.3 | 0.3 | 2.2 | 3.4 |
| Inter | 55.5 | 1777 | 206 | 646 | 127 |
| Inter-Score | 2.5 | 0 | 0.3 | 0.5 | 2.1 |

et al., 2015]) which have been suggested in the literature. The figures show illustrative results for the Karate and Dolphin networks [Zachary, 1977; Lusseau, 2003]. Objectives plotted on the top row are to be maximised, whereas those in the bottom two rows should be minimised. In each panel, the objective value corresponding to the true partition $f(P^*)$ is plotted with a green asterisk. Partitions $P$ for which the objective is misleading because $f(P) < f(P^*)$ for minimisation or $f(P) > f(P^*)$ for maximisation are shown in red. Clearly, a good objective function acts as a proxy for $NMI(P, P^*)$ and should, therefore, be well correlated with $NMI(P, P^*)$ and should not be misleading. As the figures show the majority of objective functions are quite well correlated with $NMI(P, P^*)$, but we note that the correlation in all cases is imperfect so that optimising $f(P)$ does not necessarily find the best partition. The "Inter" objective function [Shi et al., 2012] is particularly poorly correlated with $NMI(P, P^*)$ and it appears that on average maximising the inter score will lead to partitions closer to $P^*$, although its proposers suggest minimising it.

In fact, all the objective functions are misleading when they are evaluated on six real networks because there are partitions for which the objective function score is better than the score for the correct partition.

Table 3.1 summarises the number of misleading partitions found for each objective function six real-world networks. As the table shows, none of the scoring functions is completely reliable on all of the networks evaluated.

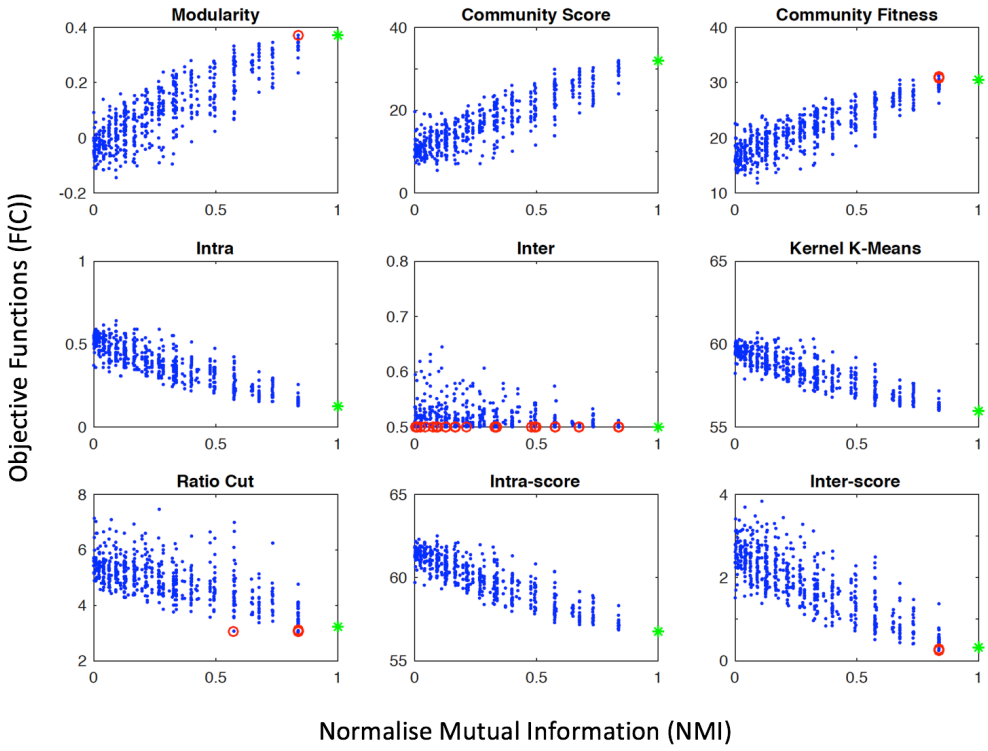


Figure 3.1 Objective function fidelity on Karate club network. Correlations of community scoring functions with $NMI$ to the true partition $P^*$. The $NMI$ between a randomly generated partition $P$ and $P^*$ is plotted horizontally versus the scoring function $f(P)$ plotted vertically. Partitions for which the scoring function is misleading are shown in red, and $f(P^*)$ is shown in green.

As Figures 3.1, 3.2 and Table 3.1 show, the proposed *Intra-Score* (equation 3.1), is generally well correlated with the $NMI$ and yields relatively few misleading partitions. The proposed *Inter-Score* (3.3) which focuses on the maximum (rather than the average) number of inter-community connections is also generally well-correlated with the $NMI$. These two scores evaluate different aspects of a candidate community, and we, therefore, seek to find good communities by simultaneously optimising both scores using a multi-objective evolutionary algorithm.

One of the limitations of our strategy is that it does not consider the reliability of ground-truth partitions. It could be that the generated partitions are better than ground-truth partitions in terms of some good structure. Although the ground truth partitions have not always strong communities but in general the structures of ground truth partitions are resemble to the natural network partitions. In addition, alternative similarity measures (see section 2.6) could be used to measure the similarity between generated partitions and the true partition. However, we used $NMI$

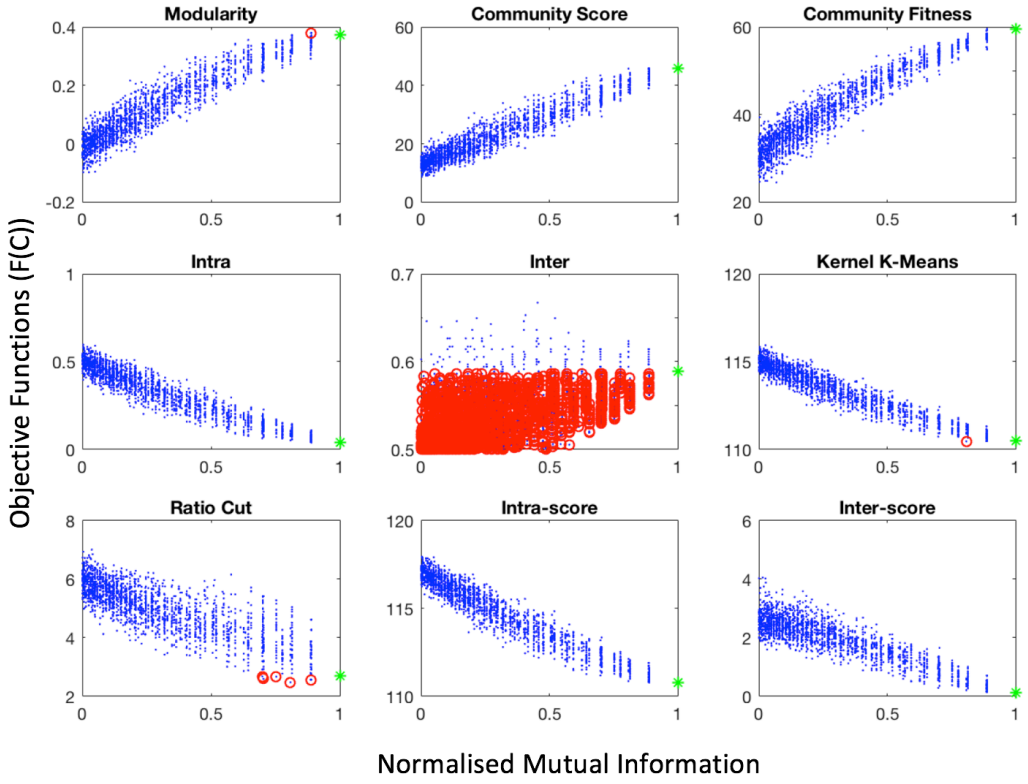


Figure 3.2 **Objective function fidelity on Dolphin network.** Correlations of community scoring functions with $NMI$ to the true partition $P^*$. The $NMI$ between a randomly generated partition $P$ and $P^*$ is plotted horizontally versus the scoring function $f(P)$ plotted vertically. Partitions for which the scoring function is misleading are shown in red, and $f(P^*)$ is shown in green.

as it is a reliable measure to find the similarity between two different community structures.

## 3.3 The proposed MOEA-CD for Community Detection

This section introduces our novel evolutionary algorithm for community detection, which we call MOEA-CD. Firstly, we briefly describe the MOEA/D algorithm, which has been shown to be successful for a wide variety of multi-objective optimisation problems [Zhang and Li, 2007]. The representation of communities in the algorithm is crucial for its efficient operation, and we describe the genotype encoding together with the genetic and local heuristic operators that promote diversity in the evolutionary population and allow exploitation of promising solutions.

A popular and robust multi-objective evolutionary algorithm is MOEA/D [Zhang

and Li, 2007] which we adapt to community detection. The cornerstone of MOEA/D is to decompose the multi-objective problem into some distinct scalar sub-problems using the Tchebycheff distance function in which the weighted objectives are linearly combined. Each sub-problem is a single objective optimisation, and it corresponds to an individual solution in an evolutionary population. All these sub-problems are optimised simultaneously with different weight vectors. The vector of weights for each of the $N_{pop}$ sub-problems is denoted by $\boldsymbol{\lambda}^j = (\lambda_1^j, \lambda_2^j, ..., \lambda_m^j)$ for $1 \leq j \leq N_{pop}$; the weights are chosen to be integer multiples of $1/N_{pop}$ and to satisfy $\sum_{i=1}^{m} \lambda_i^j = 1$. For the two objective problems that we consider here $\boldsymbol{\lambda}^j = (j/N_{pop}, 1 - j/N_{pop})$. With these weight vectors the $N_{pop}$ scalar sub-problems are defined as:

$$g_j(\mathcal{C}_j|\boldsymbol{\lambda}^j, \mathbf{z}^*) = \min_{1 \leqslant i \leqslant m} \{|\lambda_i^j f_i(\mathcal{C}_j) - z_i^*|\} \tag{3.4}$$

where $\mathbf{z}^* = (z_1, z_2, \ldots, z_m)$ is the reference point which represents the optimal value generated so far for each objective: $z_i^* = \min f_i(\mathcal{C})$.

At each generation of the evolutionary optimisation, each of the solutions $\mathcal{C}_j$ is combined with another solution chosen from its neighbours using the genetic crossover. Here the neighbours of $\mathcal{C}_j$ are defined to be the solutions whose weight vectors, $\boldsymbol{\lambda}^k$ are closest to $\boldsymbol{\lambda}^j$ using the Euclidean distance. The products of the crossover may then be mutated, after which the best solution for sub-problem $j$ is selected using dominance from the crossed-over and mutated solutions for sub-problem $j$ and its neighbours. In this way the population of sub-problem solutions $\{\mathcal{C}_j\}_{j=1}^{N_{pop}}$ can only move towards the Pareto front [Zhang and Li, 2007].

At the end of the procedure, the set of sub-problem solutions $\{\mathcal{C}_j\}_{j=1}^{N_{pop}}$ is an estimate of the Pareto set. These solutions represent a variety of partitions of the network which trade-off the two objectives.

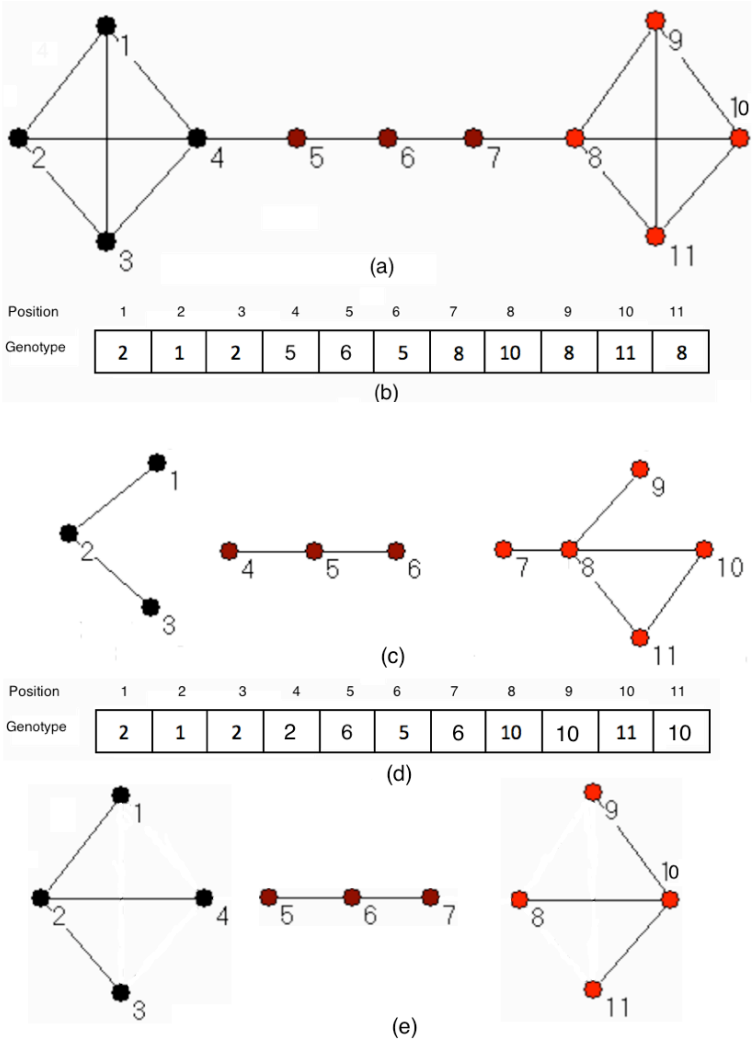


Figure 3.3 Genetic representation. (a) A simple graph with communities indicated by node colours. (b) The community structure induced by the given locus-based genetic representation, $\mathcal{C}_j = (g_1^j, g_2^j, \ldots, g_N^j)$. Here each $g_i^j$ is initialised to one of the neighbours of node $i$ [Pizzuti, 2012]. (c) The community structure resulting from Pizzuti modified initialisation. (d) Genotype induced by the given locus-based genetic representation. Here our modified initialisation in which all the unassigned neighbours of $i$ are assigned the same $g_i^j$, so that they are all in the same community. (e) The community structure by our modification.

### 3.3.1 Genetic Representation

The chromosome representation has a vital role in the efficiency of EAs. The proposed algorithm adopts the locus-based adjacency representation which has been proposed by Park and Song in 1998 for genotype encoding [Park and Song, 1998]. This representation has been employed by Handl and Knowles [2007] for multi-objective clustering and is commonly used by evolutionary algorithms for community detection [Pizzuti, 2008, 2012; Shi et al., 2012; Gong et al., 2014; Hafez et al., 2014; Wu and Pan, 2015]. In this method, each individual $\mathcal{C}_j$ corresponds to a net-

**Algorithm 3.2** Individual initialisation.

Inputs
1 : $A$ : Adjacency matrix
2 : $N$ : Number of nodes in the network
3 : $\mathcal{C}$ : Individual (partition)
Steps
1 : $\mathcal{C} = (g_1, g_2, \ldots, g_N) \leftarrow 0$ ▷ individual consists of a number of genes.
2 : **for** $v = 1$ **to** $N$ **do** ▷ Each node $v$ corresponds to one gene
3: **if** $g_v == 0)$ **then**
4: $u \leftarrow random(Neighbor(v))$
5 : **for** $i = v$ **to** $N$ **do**
6: **if** $g_i == 0)\&\&(A(i,u) == 1)$**then**
7: $g_i \leftarrow u$ ▷ spread random neighbor as allel to all neighbors.
8: **end if**
9: **end for**
10: **end if**
11: **end for**
12: return($\mathcal{C}$) ▷ Individual

work partition and consists of $N$ genes, $\mathcal{C}_j = (g_1^j, g_2^j, \ldots, g_N^j)$ where $N$ is the number of nodes in the network. Each gene corresponds to a node in the network, and $g_i^j$ indicates that node $i$ and node $j$ belong to the same community. In the initial Park and Song formulation, the $g_i^j$ were initialised randomly. This representation has the advantage that the number of communities does not have to be specified *a priori*. However, the initialisation of the $g_i^j$ to random values may often lead to nodes which are distant from the original network being assigned to the same community.

As illustrated in Figure 3.3b, Pizzuti improved the initialisation by insisting that $j$ is initialised to one of the neighbours of node $i$ [Pizzuti, 2012]. Intuitively we expect a node to have links to other nodes in the same community. Here we therefore further bias the initialisation towards strong nodes in strong communities by initialising to the same community all the neighbours of $i$ which have not yet been assigned communities. Figure 3.3d illustrates that the allele (node 2) for gene 1 (node 1) is chosen randomly from its neighbours and the same allele (node 2) is given to all genes that have connections with node 2 such as genes 3 and 4, as illustrated in Algorithm 3.2.

**Algorithm 3.3** Local heuristic search (Neighbour Node Centrality algorithm).

Inputs
1 : $d$ : The degree of each node in the network
2 : $N$ : Number of nodes in the network
3 : $\mathcal{C}$ : Individual (partition)
3 : $p_m$ : The mutation probability
Steps
1 : $\mathcal{C} = \{C_1, C_2, \ldots, C_K\}$ ▷ $K$ Maximum number of communities in $\mathcal{C}$
2 : **for** $v = 1$ **to** $N$ **do**
3 : **if**$(d(v)) > 0)\&\&(rand \leq pm)$ **then**
4 : $C_v \leftarrow Community(v)$ ▷ $C_v$ is the community of the node $v$
5 : **if**$(\underline{d}(v, C_v)) \leq \overline{d}(v, C_v))$ **then** ▷ Node $v$ is a weak node
6 : $u^* \leftarrow \arg\max_{u \sim v} d(u)$
7 : $Community(v) \leftarrow Community(u^*)$
8: **end if**
9: **end if**
10: **end for**
11: return($\mathcal{C}$)

### 3.3.2 Genetic and Neighbour Node Centrality Operators.

Before we introduce a novel local heuristic search based mutation operator, we describe briefly the crossover operator used in the algorithm. In general, a crossover operator combines the features from two chromosomes to generate offspring. Here we use a standard uniform crossover operator, in which each gene of the offspring is selected from one parent with probability $p_c$ and from the other with probability $1 - p_c$; here $p_c = \frac{1}{2}$. This operator is adopted because, in conjunction with this genetic representation, it avoids generating worthless solutions in which a node is completely disconnected [Pizzuti, 2012]. A single offspring is generated for each sub-problem, where the parents for the offspring are chosen at random from the five nearest neighbours to the sub-problem.

A standard mutation strategy used by many authors [Shi et al., 2012; Pizzuti, 2012; Hafez et al., 2014] is as follows. Each gene in each chromosome is mutated with probability $p_m$ by changing the node to which it is connected in the genetic representation, which determines the node's community, to a randomly chosen neighbouring node. We propose an alternative mutation procedure based on *Neighbour Node Centrality*, which we show in section 3.4 aids convergence. If the node to be mutated is a strong

node (the node that has internal connections more than external connections, see Equation 2.10), then its community is left unchanged. On the other hand, if it is a weak node (the node that has internal connections less than external connections, see Equation 2.11), then its community is set to be the community of its neighbour with the most connections. Specifically, if $v$ is the weak node to be mutated, then let $u^* = \arg\max_{u \sim v} d(u)$ be the central neighbour of $v$, where $u \sim v$ indicates that $u$ and $v$ are neighbours. Then the community of $v$ is assigned to be the community of $u^*$, as illustrated in Algorithm 3.3.

The time complexity per generation of the algorithm is dominated by the time taken to evaluate the objectives, which takes $O(N^2)$ time. At each generation $N_{pop}$ new solutions must be evaluated, so the overall worst case complexity per generation is $O(N^2 \times N_{pop})$.

## 3.4 Experiments

In this section, we present and discuss the results which show the effectiveness of the proposed MOEA-CD algorithm compared to three state-of-the-art methods, namely, MOGA-Net [Pizzuti, 2012], MOCD [Shi et al., 2012] and MODPSO [Gong et al., 2014]. Here the name of these algorithms refers to the authors' objectives and not the authors' algorithms. In order to evaluate the efficacy of our new objectives and to provide a fair comparison, we used our algorithm for optimising the objectives defined by each of these authors rather than re-implementing their entire algorithms. The methods are evaluated on 28 networks, which are classified into three groups: The first group contains the LFR benchmark networks [Lancichinetti et al., 2008]. The second group comprises five real-world networks for which the ground-truth partitions are known. These networks are the Zachary karate club network[1] [Zachary, 1977], the Bottlenose Dolphin network[2] [Lusseau, 2003], the

[1] http://networkdata.ics.uci.edu/data/karate/
[2] http://networkdata.ics.uci.edu/data/dolphins/

Table 3.2 Network characteristics.

| Networks | Nodes | Edges | Clusters |
|---|---|---|---|
| Karate | 34 | 78 | 2 |
| Dolphin | 62 | 159 | 2 |
| Krebs' books | 105 | 440 | 3 |
| Football2000 | 115 | 613 | 12 |
| Football2001 | 115 | 613 | 19 |
| SFI | 118 | 200 | unknown |
| Jazz | 198 | 2742 | unknown |
| Netscience | 1589 | 2742 | unknown |

American football network[3] [Girvan and Newman, 2002], and the Krebs' American politics network[4] [Newman, 2006]. Finally, the third group comprises three real-networks for which the ground-truth partitions are unknown. These networks are the Santa Fe Institute (SFI) network[5] [Girvan and Newman, 2002], the Jazz Musician network[6] [Gleiser and Danon, 2003]) and the Netscience network[7] [Newman, 2006]. Characteristics of the real-world networks are given in Table 3.2.

We used the MOEA/D algorithm with the following parameters. Both the number of sub-problems (population size) and the number of generations were 300, the neighbourhood size was 5, and the cross-over probability was $p_c = 0.8$. The mutation probability $p_m = 0.6$. The mutation rate is larger than usually used, however preliminary investigations showed that this higher rate was beneficial. Higher mutation rates do not provide any further benefit.

In order to evaluate the algorithms, we calculated the maximum $NMI$ between the true partition $P^*$ and the union of all partitions forming the Pareto fronts of twenty runs of the algorithm together with average (over the twenty runs) of the maximum $NMI$ between the correct partition and partitions in a Pareto front from one run. These are denoted $NMI_{\max}$ and $NMI_{av}$ respectively. We also evaluate the maximum and average modularity over the twenty runs, denoted by $Q_{max}$ and $Q_{av}$ respectively.

[3] http://networkdata.ics.uci.edu/data/football/
[4] http://networkdata.ics.uci.edu/data/polbooks/
[5] http://dsec.pku.edu.cn/~jliu/
[6] http://konect.uni-koblenz.de/networks/arenas-jazz
[7] http://vlado.fmf.uni-lj.si/pub/networks/data/collab/netscience.htm

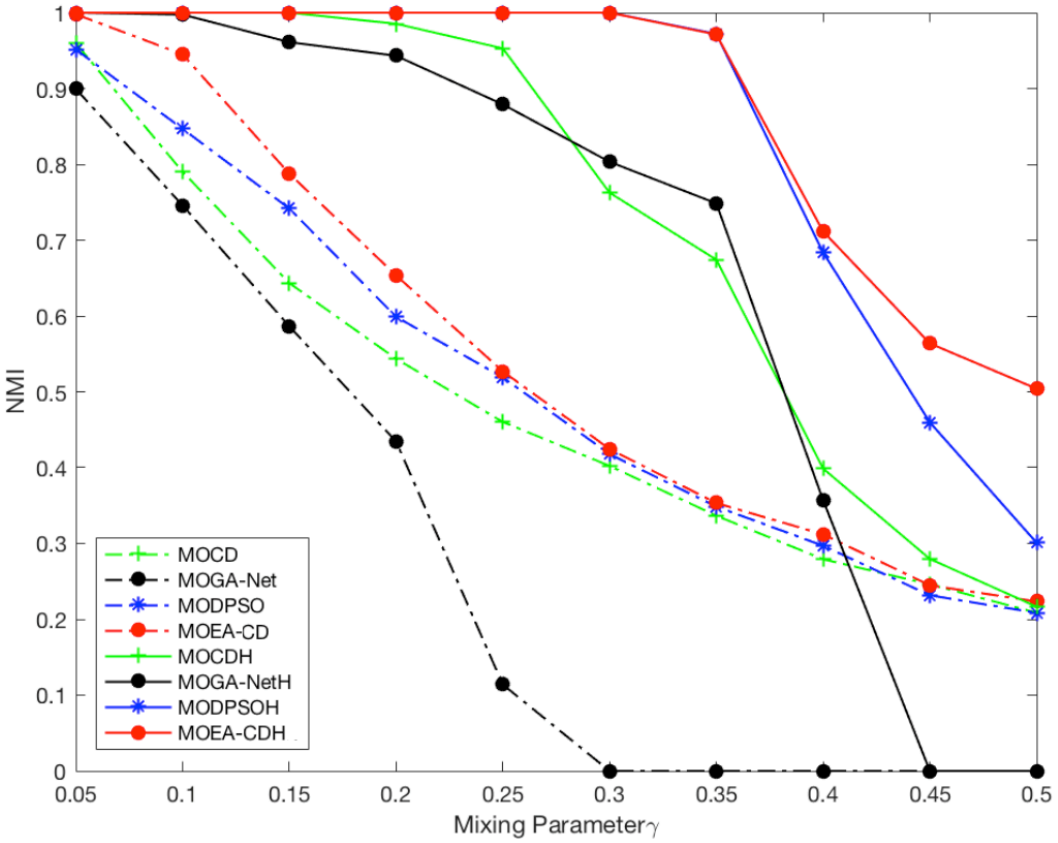


Figure 3.4 Average best $NMI$ between ground truth and detected partitions for MOGA-Net (black), MOCD (green), MODPSO (blue) and MOEA-CD (red) over twenty runs on the LFR128 benchmark networks (10 networks) with and without the Neighbourhood Node Centrality heuristic. Dashed and solid lines indicate results without and with the heuristic respectively.

### 3.4.1 Synthetic Networks.

In our first set of experiments, we test all models on computer-generated benchmark networks. These benchmark networks where proposed by Girvan and Newman [2002] and extended by Lancichinetti et al. [2008].

In the first group of the LFR benchmarks, each network has 128 nodes, and each is constructed to contain 4 communities of 32 nodes. The extent of connections between communities is controlled by the mixing parameter $\gamma$, which is the probability that a node has an edge to a node outside its community. Thus when $\gamma$ is small, the community structure is strong and diminishes as $\gamma$ increases. Here we compare the performance of the algorithms on networks generated with $\gamma$ in the range $[0.05, 0.5]$. We denote these networks by LFR128.

Figure 3.4 shows summary results for the performance of all models in terms of the average $NMI$. Dashed lines indicate results for each algorithm without the Neighbourhood Node Centrality heuristic and solid lines show $NMI$ for algorithms using the Neighbourhood Node Centrality heuristic. Unsurprisingly, all algorithms tend to do better when the community structure is strong ($\gamma$ small). As the mixing parameter increases, the performance drops as the communities become less distinct. However, in all cases, the addition of the Neighbourhood Node Centrality mutation heuristic substantially enhances the performance because it focuses on constructing strong communities. Also, the proposed MOEA-CD algorithm shows superior performance, particularly for large $\gamma$. The MODPSOH model ([Gong et al., 2014] with Neighbourhood Node Centrality) also performs well. This method optimises the ratio cut and kernel $k$-means objectives which, as shown above (Figures 3.1 and 3.2), perform well for strong communities.

The second group of the LFR benchmarks is used to test the four models with larger size networks which are similar to real-world networks. These benchmarks comprise 10 networks, each one consisting of 1000 nodes, and we, therefore, denote this group as LFR1000. The degree and community size distributions of these networks obey power laws with exponents 2 and 1 respectively [Lancichinetti et al., 2009]. As before, networks are generated with different mixing parameters $\gamma$, which controls the probability of an edge making an inter-community connection. We tested our algorithm on networks with $\gamma$ ranging from 0.05 to 0.5 in steps of 0.05.

Figure 3.5 summarises the $NMI_{av}$ over twenty runs for the four models on the LFR1000 benchmark datasets. The lower four dashed lines represent the results for these models without the Neighbourhood Node Centrality heuristic, and it is clear that employing this method enhances the evolutionary search, allowing better partitions to be found. With the Neighbourhood Node Centrality heuristic, all algorithms except MOGA-Net perform well, yielding Pareto fronts which contain partitions close to the correct partition.

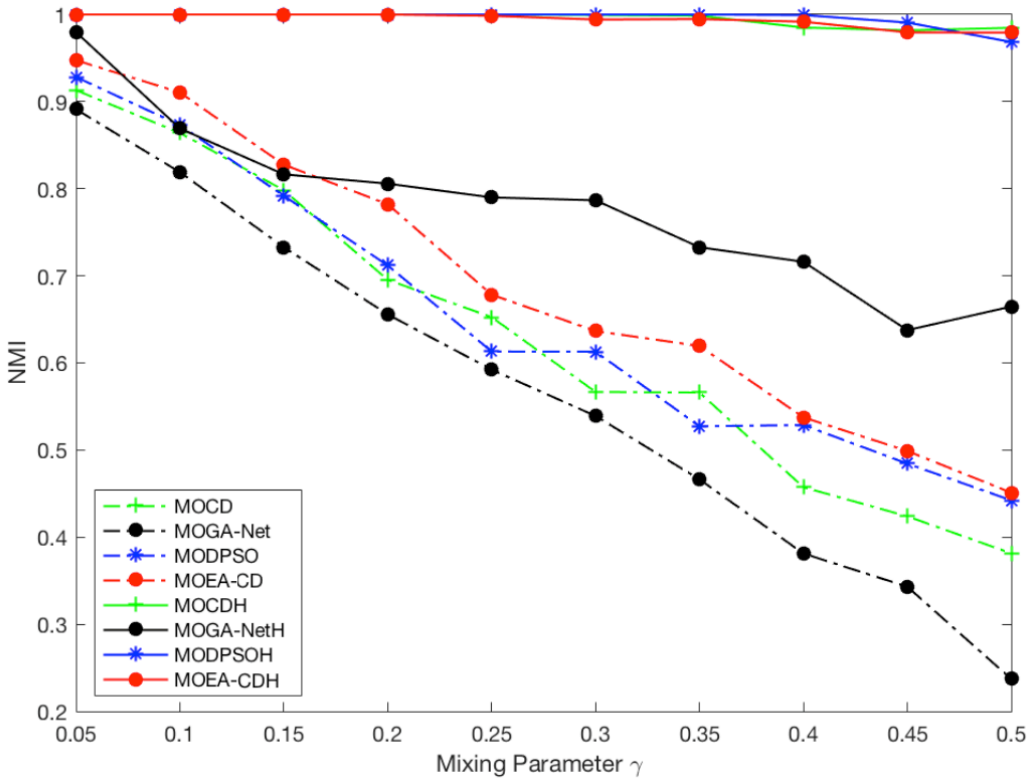


Figure 3.5 Average best $NMI$ between ground truth and detected partitions for MOGA-Net (black), MOCD (green), MODPSO (blue) and MOEA-CD (red) over twenty runs on the LFR1000 benchmark networks (10 networks) with and without the Neighbourhood Node Centrality heuristic. Dashed and solid lines indicate results without and with the heuristic respectively.

### 3.4.2 Real-world networks with ground-truth partitions

Tables 3.3 and 3.4 compare the performance of the MOGA-Net, MOCD, MODPSO and MOEA-CD algorithms on 5 real-world networks for which the correct partition is known. Tables 3.3 shows the results without heuristic ($NNC$) while 3.4 shows the results with heuristic ($NNC$).

Table 3.3 reports the statistical results of four models over twenty different runs on five real-world networks whose correct partitions are known without using *Neighbourhood Node Centrality* procedure. The bold number refers to the detected partition which most resembles the true partition. Here we used two evaluation scores: $NMI$ and $Q$.

First, we start with the Zachary's Karate Club network [Zachary, 1977]; it is the

Table 3.3 Maximum and average of $NMI$ and modularity for testing four models without Neighbourhood Node Centrality on five real-world networks whose the ground-truth partition is known. $NMI_{Qmax}$ measures the similarity between $Q_{max}$ and true partition for each network. $POS_{av}$ is the average size of the Pareto optimal sets which have been generated by different algorithms over twenty independent runs. $POS_{min}$ and $POS_{max}$ are the smallest and the largest values among the approximation sets for each algorithm on each network respectively. The best score achieved for each network is in bold font.

| Networks | Criteria | MOCD | MOGA-Net | MODPSO | MOEA-CD |
|---|---|---|---|---|---|
| Karate | $NMI_{max}$ | **0.8372** | **0.8372** | **0.8372** | **0.8372** |
| | $NMI_{av}$ | 0.8370 | 0.8065 | **0.8371** | **0.8371** |
| | $Q_{max}$ | 0.4087 | 0.4018 | **0.4188** | **0.4188** |
| | $NMI_{Qmax}$ | 0.5305 | **0.6317** | 0.5866 | 0.5866 |
| | $Q_{av}$ | 0.3952 | 0.3832 | 0.4092 | **0.5866** |
| | $POS_{av}$ | 30 | 7.5 | 13.4 | 11.7 |
| | $POS_{min}$ | 22 | 5 | 10 | 8 |
| | $POS_{max}$ | 35 | 12 | 18 | 18 |
| Dolphin | $NMI_{max}$ | **1** | 0.88888 | **1** | **1** |
| | $NMI_{av}$ | 0.9532 | 0.8125 | 0.9778 | **1** |
| | $Q_{max}$ | 0.4674 | 0.4675 | **0.5199** | 0.48742 |
| | $NMI_{Qmax}$ | 0.5338 | **0.5980** | 0.5821 | 0.5909 |
| | $Q_{av}$ | 0.4578 | 0.4440 | **0.4800** | 0.4719 |
| | $POS_{av}$ | 54 | 9.3 | 41.9 | 43.2 |
| | $POS_{min}$ | 45 | 5 | 29 | 32 |
| | $POS_{max}$ | 66 | 14 | 51 | 52 |
| Football 2000 | $NMI_{max}$ | 0.7224 | 0.5433 | 0.7814 | **0.8803** |
| | $NMI_{av}$ | 0.7029 | 0.6207 | 0.7291 | **0.7625** |
| | $Q_{max}$ | 0.4356 | 0.4206 | 0.4666 | **0.5490** |
| | $NMI_{Qmax}$ | 0.6718 | 0.5980 | 0.6163 | **0.8803** |
| | $Q_{av}$ | 0.4104 | 0.3898 | 0.4212 | **0.4566** |
| | $POS_{av}$ | 68.5 | 10.7 | 22 | 21.9 |
| | $POS_{min}$ | 56 | 7 | 16 | 12 |
| | $POS_{max}$ | 91 | 14 | 26 | 27 |
| Football 2001 | $NMI_{max}$ | 0.7550 | 0.7252 | 0.8111 | **0.8367** |
| | $NMI_{av}$ | 0.7390 | 0.6680 | 0.7890 | **0.8102** |
| | $Q_{max}$ | 0.4300 | 0.4177 | 0.4708 | **0.4869** |
| | $NMI_{Qmax}$ | 0.7070 | 0.5807 | **0.7391** | 0.6809 |
| | $Q_{av}$ | 0.4090 | 0.3810 | 0.4328 | **0.4611** |
| | $POS_{av}$ | 68.4 | 10.1 | 22.4 | 24 |
| | $POS_{min}$ | 57 | 8 | 19 | 15 |
| | $POS_{max}$ | 85 | 15 | 26 | 31 |
| Krebs' | $NMI_{max}$ | 0.6656 | 0.6042 | 0.6947 | **0.7331** |
| | $NMI_{av}$ | **0.6174** | 0.5307 | 0.6040 | 0.5975 |
| | $Q_{max}$ | 0.5107 | 0.4806 | **0.5190** | 0.5165 |
| | $NMI_{Qmax}$ | 0.5629 | **0.6042** | 0.5671 | 0.5866 |
| | $Q_{av}$ | 0.4882 | 0.4655 | **0.5014** | 0.4963 |
| | $POS_{av}$ | 63.8 | 7.6 | 48.6 | 30.8 |
| | $POS_{min}$ | 52 | 4 | 41 | 21 |
| | $POS_{max}$ | 73 | 11 | 60 | 41 |

most popular network which has been used as a benchmark to evaluate community detection algorithms. In this study, Zachary observed 34 club members over a period of two years in the United States. Due to a conflict between the club administrator (node 34) and the instructor (node 1), the club was separated into small communities. Nodes 3 and 10 fall between two communities. These nodes are usually represented as local optima by most community detection algorithms. Table 3.3 indicates that all models misclassify either node 3 or 10 when they are tested without using the $NNC$ strategy. Thus, the produced $NMI$ values of all models are quite close ($NMI = 0.8372$). The larger $Q_{max}$ and $Q_{av}$ values mean that the algorithm has a good convergence capability. $NMI_{Qmax}$ measures how good the single partition that would be selected from the Pareto set on the basis of maximising Q over 20 runs.

Table 3.3 also shows the size of the approximations to the Pareto optimal solutions. $POS_{av}$ represents the average number of the Pareto optimal solutions that have been generated by four different algorithms over twenty runs for each network. The $POS_{min}$ and $POS_{max}$ are the smallest and the largest values among the approximation sets for each algorithm respectively. The results show that the MOCD algorithm has the largest number of Pareto optimal solutions among the other algorithms while MOGA-Net algorithm produces the smallest number. MODPSO and our algorithms have similar numbers of Pareto optimal solutions. Therefore MOCD algorithm has the more opportunity to choose the $NMI_{max}$ among a large number of solutions compare with other algorithms.

The second real-world network represents the social interaction of Bottlenose Dolphins living in Doubtful, New Zealand, over a period of seven years. It was compiled by Lusseau [2003]; nodes represent dolphins and links represent frequent associations between dolphin pairs. On this network, only our algorithm (MOEA-CD) converges to the global optimum, and it can figure out the correct partition at $NMI = 1$ while all the remaining algorithms are trapped at different local optima.

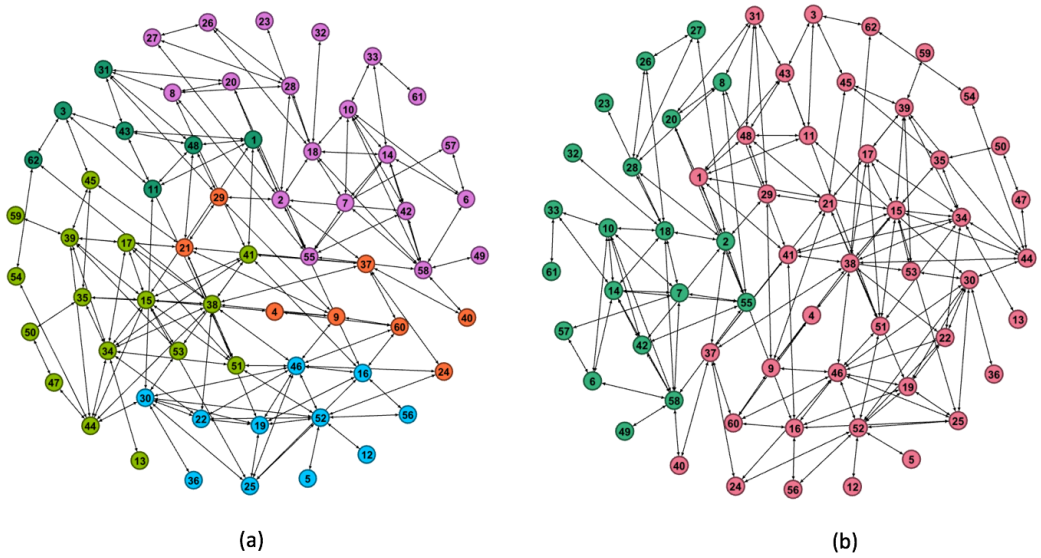


Figure 3.6 Dolphin network partition without heuristic. (a) Community structure obtained by MODPSO. This partition corresponds to the partition that have maximum modularity ($Q_{max} = 0.5199$) and $NMI = 0.5820$. (b) Community structure obtained by MOEA-CD. This partition corresponds to the partition that have maximum $NMI$ of 1 and $Q = 0.3734$.

The MODPSO model misplaces node 31 in some runs although node 31 has $\underline{d}(v_{31}, C_k) > \overline{d}(v_{31}, C_k)$ while in the other runs it finds the true partition of the network. Therefore, the $NMI$ average over twenty runs is 0.9778. Also, frequently MOGA-Net misclassifies nodes 31 and 8 among other misclassified nodes in some runs, and MOCD misclassifies node 31 in some runs. We can infer that only our model is succeeding in classifying these nodes correctly by minimising the maximum node external connections with respect to the connections inside the community for that node. Although our algorithm can detect the true partition, the best value of modularity $Q_{max}$ and $Q_{av}$ is for MODPSO. The $NMI$ value for the partition that has maximum modularity is 0.5821. This partition is illustrated in Figure 3.6a in five communities. On the other hand, the modularity value for the partition that has a maximum $NMI$ of 1 is 0.3734. Figure 3.6b corresponds to the true partitioning of this network into two strong communities.

It is worth noting that the larger value of modularity does not always correspond to the best partition over Pareto-optimal solutions. Maximising modularity has limitations in community detection as it tends to split large communities when the resolution is high [Lancichinetti and Fortunato, 2011]. We have seen earlier in

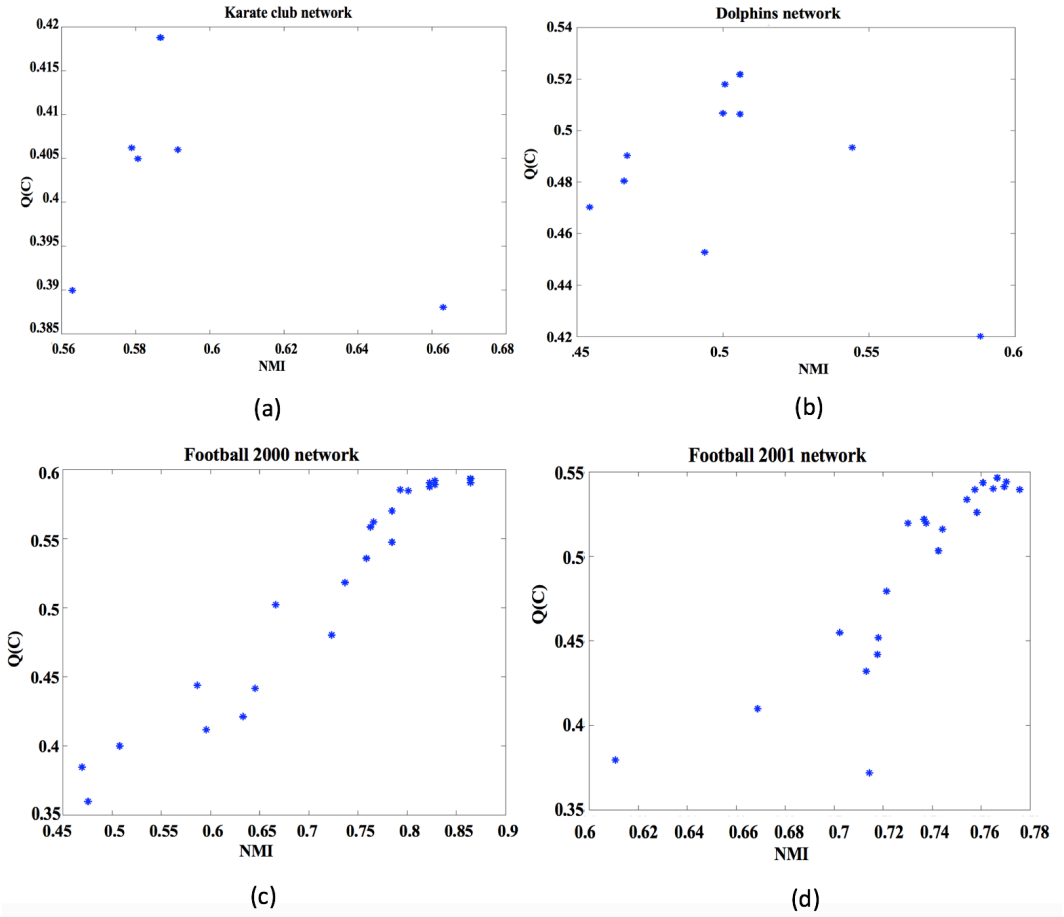


Figure 3.7 Correlations of modularity for the partition that has maximum Q at each generation with $NMI$ to the true partition. The $NMI$ between the true partition $P^*$ and the partition $P$ that has maximum modularity is plotted horizontally versus the modularity $Q$ plotted vertically. SOEA is used to optimise modularity and produce partitions $P$ without Neighbourhood Node Centrality. (a) Modularity evaluation on the Karate club networks (b) Modularity evaluation on the Dolphins networks. (c) Modularity evaluation on the Football 2000 networks. (d) Modularity evaluation on the Football 2001 networks.

section 3.1 that Q is misleading in the Karate and Dolphins network partitions. In addition to that, we optimise modularity using a Single Objective Evolutionary Algorithm to show the correlation between modularity and $NMI$ where $NMI$ is calculated between the true partition and the partitions that are obtained by the single evolutionary algorithm as illustrated in Figures 3.7 and 3.8. We conclude from these two figures that the modularity has many misleading points on the Karate and Dolphin networks while it looks has a good behaviour on the Football networks.

Now, Table 3.4 shows the effect of applying the $NNC$ method through mutation operator on the performance of these four models to detect the structure of communities on real-world networks. In general, there is a clear improvement in the performance of all models gained by using the $NNC$ method. For the Dolphin net-

Table 3.4 Maximum and average of $NMI$ and modularity for testing four models with Neighbourhood Node Centrality on five real-world networks whose the ground-truth partition is known. $NMI_{Qmax}$ measures the similarity between $Q_{max}$ and true partition for each network. $POS_{av}$ is the average size of the Pareto optimal sets which have been generated by different algorithms over twenty independent runs. $POS_{min}$ and $POS_{max}$ are the smallest and the largest values among the approximation sets for each algorithm on each network respectively. The best score achieved for each network is in bold font.

| Networks | Criteria | MOCD | MOGA-Net | MODPSO | MOEA-CD |
|---|---|---|---|---|---|
| Karate | $NMI_{max}$ | 0.8822 | **1** | **1** | **1** |
| | $NMI_{av}$ | 0.8372 | **1** | **1** | **1** |
| | $Q_{max}$ | **0.4198** | **0.4198** | **0.4198** | **0.4198** |
| | $NMI_{Qmax}$ | **0.6873** | **0.6873** | **0.6873** | **0.6873** |
| | $Q_{av}$ | 0.4141 | 0.4156 | **0.5014** | 0.4142 |
| | $POS_{av}$ | 34 | 11.4 | 19.8 | 19 |
| | $POS_{min}$ | 28 | 11 | 17 | 13 |
| | $POS_{max}$ | 46 | 12 | 26 | 26 |
| Dolphin | $NMI_{max}$ | **1** | **1** | **1** | **1** |
| | $NMI_{av}$ | 0.8941 | **1** | **1** | **1** |
| | $Q_{max}$ | **0.5277** | **0.5277** | 0.5263 | 0.5268 |
| | $NMI_{Qmax}$ | 0.5932 | 0.5932 | **0.6363** | 0.5715 |
| | $Q_{av}$ | **0.5255** | 0.5216 | 0.5126 | 0.5189 |
| | $POS_{av}$ | 71.6 | 29.1 | 42.2 | 30.2 |
| | $POS_{min}$ | 59 | 26 | 33 | 21 |
| | $POS_{max}$ | 84 | 37 | 63 | 45 |
| Football 2000 | $NMI_{max}$ | **0.9361** | 0.8772 | 0.9286 | 0.9315 |
| | $NMI_{av}$ | **0.9276** | 0.8523 | 0.9253 | 0.9271 |
| | $Q_{max}$ | **0.6046** | 0.5881 | 0.6043 | **0.6046** |
| | $NMI_{Qmax}$ | **0.8903** | 0.7949 | 0.8850 | **0.8903** |
| | $Q_{av}$ | **0.6037** | 0.5725 | 0.6034 | 0.6034 |
| | $POS_{av}$ | 92.4 | 22 | 23.7 | 30.3 |
| | $POS_{m}in$ | 72 | 19 | 19 | 18 |
| | $POS_{m}ax$ | 112 | 26 | 29 | 39 |
| Football 2001 | $NMI_{max}$ | **0.9757** | 0.9241 | 0.9690 | 0.9690 |
| | $NMI_{av}$ | **0.9696** | 0.9038 | 0.9686 | 0.9673 |
| | $Q_{max}$ | **0.6046** | 0.5861 | **0.6046** | **0.6046** |
| | $NMI_{Qmax}$ | **0.9328** | 0.9017 | **0.9328** | **0.9328** |
| | $Q_{av}$ | **0.6037** | 0.5725 | 0.6035 | 0.6034 |
| | $POS_{av}$ | 96.7 | 23.2 | 21.6 | 18.7 |
| | $POS_{min}$ | 78 | 20 | 17 | 14 |
| | $POS_{max}$ | 117 | 29 | 27 | 23 |
| Krebs' books | $NMI_{max}$ | **0.6776** | 0.6339 | 0.6210 | 0.6366 |
| | $NMI_{av}$ | **0.6171** | 0.5939 | 0.6064 | 0.5861 |
| | $Q_{max}$ | **0.5254** | 0.5245 | 0.5251 | 0.5247 |
| | $NMI_{Qmax}$ | 0.5405 | 0.5537 | 0.5289 | **0.5735** |
| | $Q_{av}$ | 0.5237 | 0.5215 | **0.5289** | 0.5226 |
| | $POS_{av}$ | 107.7 | 24.5 | 45 | 27.8 |
| | $POS_{min}$ | 63 | 17 | 37 | 22 |
| | $POS_{max}$ | 138 | 34 | 51 | 42 |

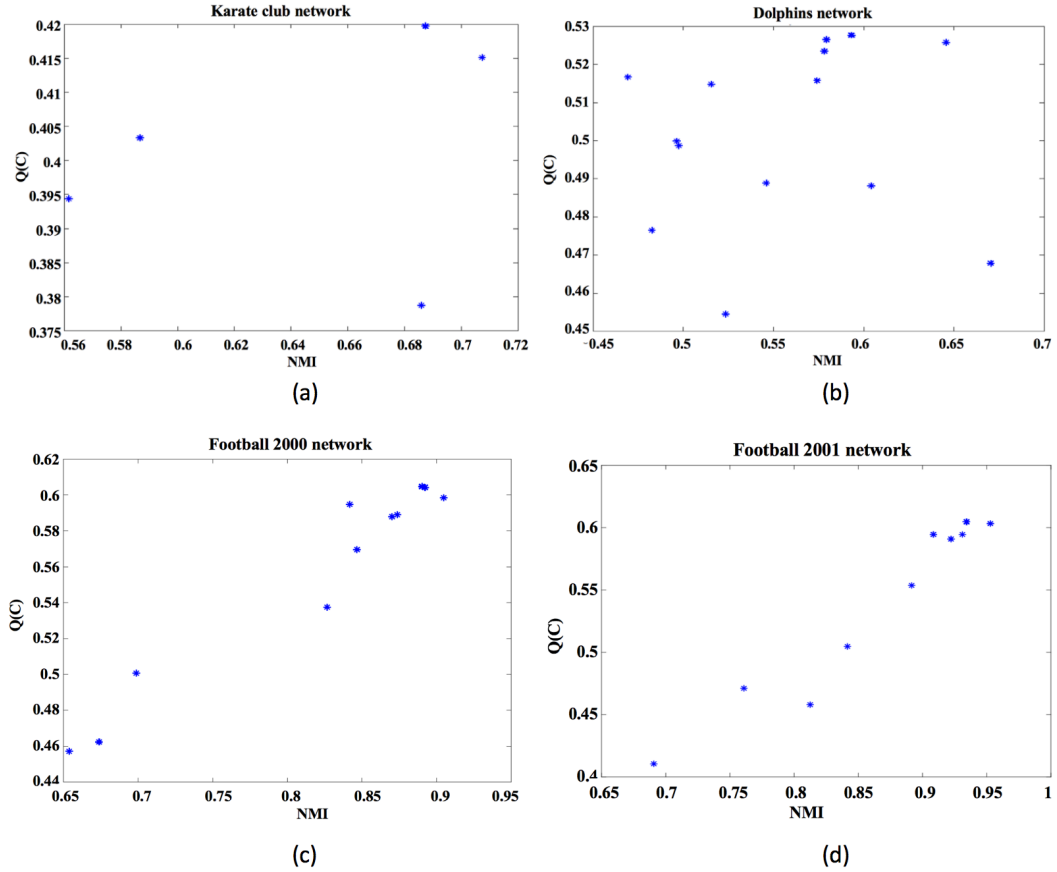


Figure 3.8 Correlations of modularity for the partition that has maximum $Q$ at each generation with $NMI$ to the true partition. The $NMI$ between the true partition $P^*$ and the partition $P$ that has maximum modularity is plotted horizontally versus the modularity $Q$ plotted vertically. SOEA is used to optimise modularity and produce partitions $P$ with Neighbourhood Node Centrality at each generation. (a) Modularity evaluation on the Karate club networks (b) Modularity evaluation on the Dolphins networks. (c) Modularity evaluation on the Football 2000 networks. (d) Modularity evaluation on the Football 2001 networks.

work, all models can reveal the true structure of communities except MOGA-Net. For the Football 2000 network, we can clearly see that there is a competition between our and MOCD models. In addition, these results show that the average number of Pareto optimal set is increased for all algorithms on most networks as there are different solutions are added due to the combined $NNC$ strategy.

Figure 3.9a shows archive solutions for our model on the Karate network in one run. Figure 3.9b displays the correct partition which is detected by MOEA-CD at $NMI = 1$. It shows the positive effect of applying $NNC$ strategy by reassigning nodes (like node 3 here) which have same connections within the community and with the rest to the community of the neighbour node that has more strong connections (like node 1 here), in the same case, node 10 is reassigned to node 34. As a result, these nodes

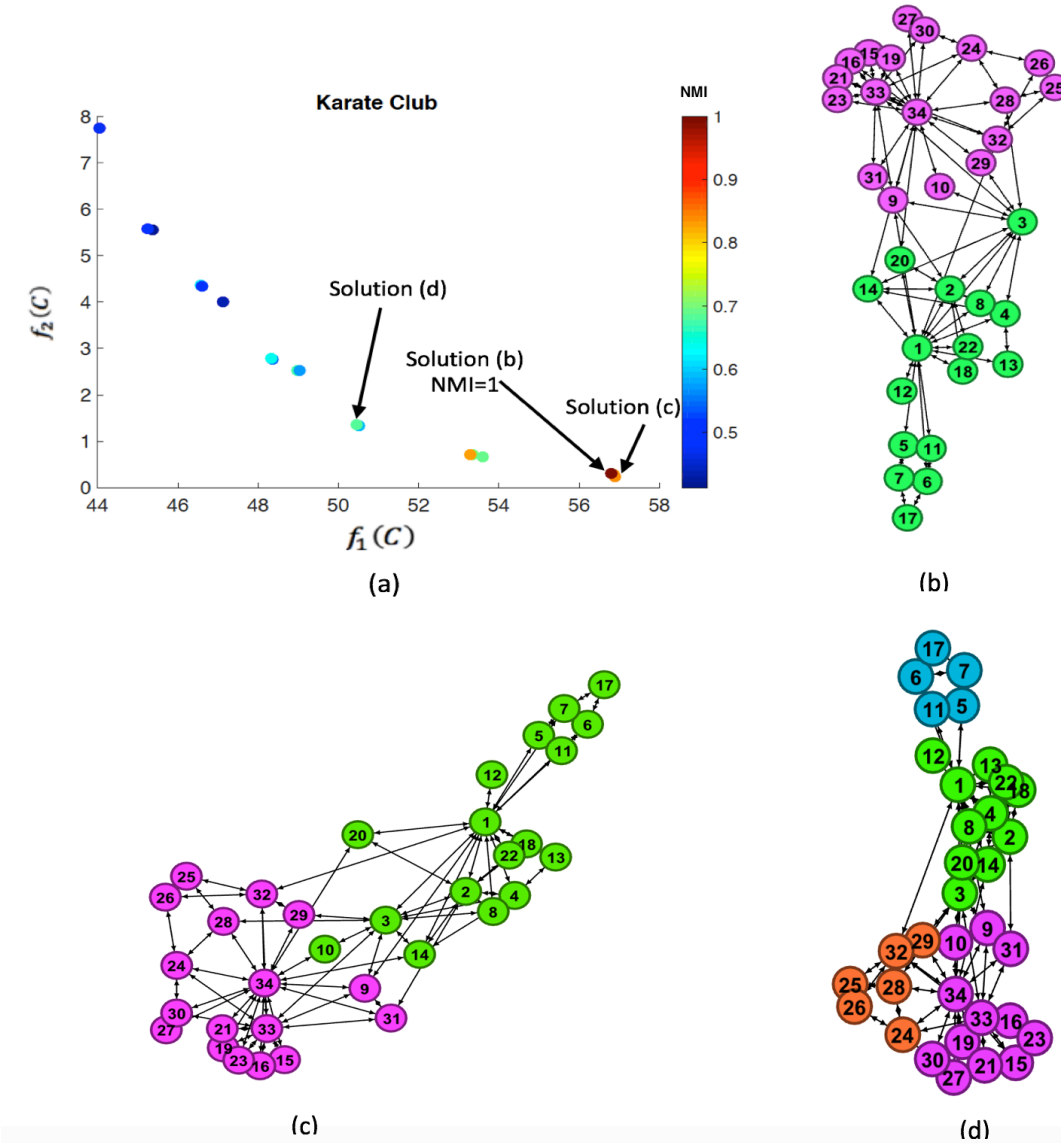


Figure 3.9 Community detection results on the karate club network by MOEA-CD model. (a) Pareto front of one run with the $NNC$ method. The colour bar represents the range of $NMI_{max}$ values. (b) Detected correct community structure which corresponds to solution b at $NMI = 1$. This is the best among a set of trade-off solutions. (c) Detected community structure which is corresponding to solution c at $NMI = 0.8371$, only node 10 is misclassified. (d) Detected community structure which is corresponding to solution d at $NMI = 0.6872$, the network is divided into four communities. Colours indicate the community that a node belongs to.

(node 3 and node 10) are assigned to the correct community. Figure 3.9c represents the partition at $NMI = 0.8372$, this partition corresponds to local optima in these models as these models misclassify node which has the same number of connections within the community and with other communities. Figure 3.9d shows the division of the network into four communities that correspond to solution d in the Pareto front plot.

Figure 3.10a displays archive solutions which are obtained in one run on the Dolphin

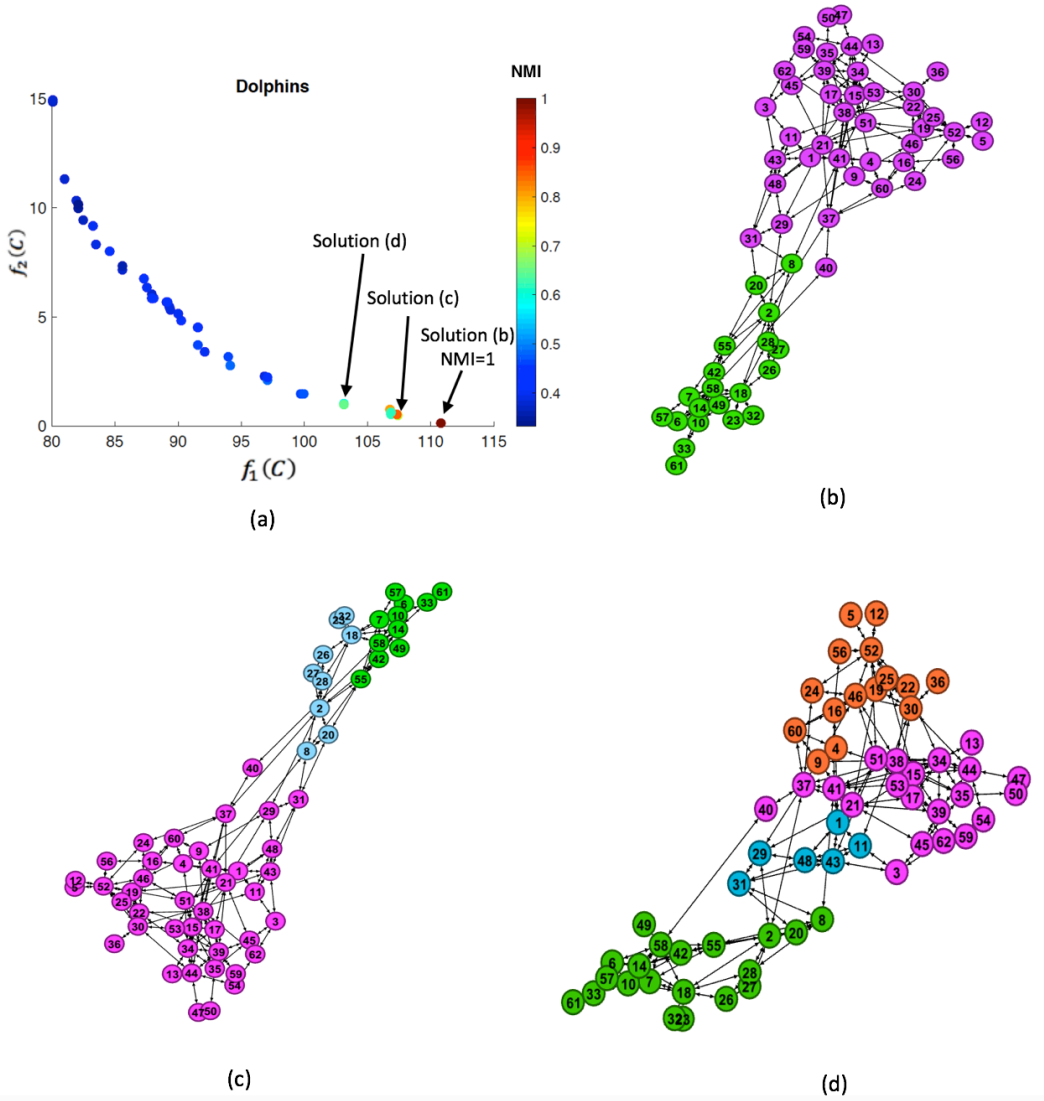


Figure 3.10 Community detection results on the Dolphin network by MOEA-CD model. (a) Pareto front of one run with $NNC$ method. (b) Detected correct community structure which is corresponding to solution b at $NMI = 1$. (c) Detected community structure which is corresponding to solution c at $NMI = 0.8499$, the network is divided into three communities. (d) Detected community structure which is corresponding to solution d at $NMI = 0.6516$, the network is divided into four communities. Colours indicate the community that a node belongs to.

network. The detected correct partition by our model is illustrated in Figure 3.10b. Figure 3.10c shows the division of the upper community of the correct partition into two communities while Figure 3.10d shows the division of the lower community of the correct partition into three communities.

On the other hand, Figure 3.11a shows that both objectives contribute to producing Pareto-optimal solutions. Thus, we conclude that SOEA can successfully detect community structures on some real-world networks while it fails on others. Note also for Football 2001, Table 3.4 shows that there is strong competition among three

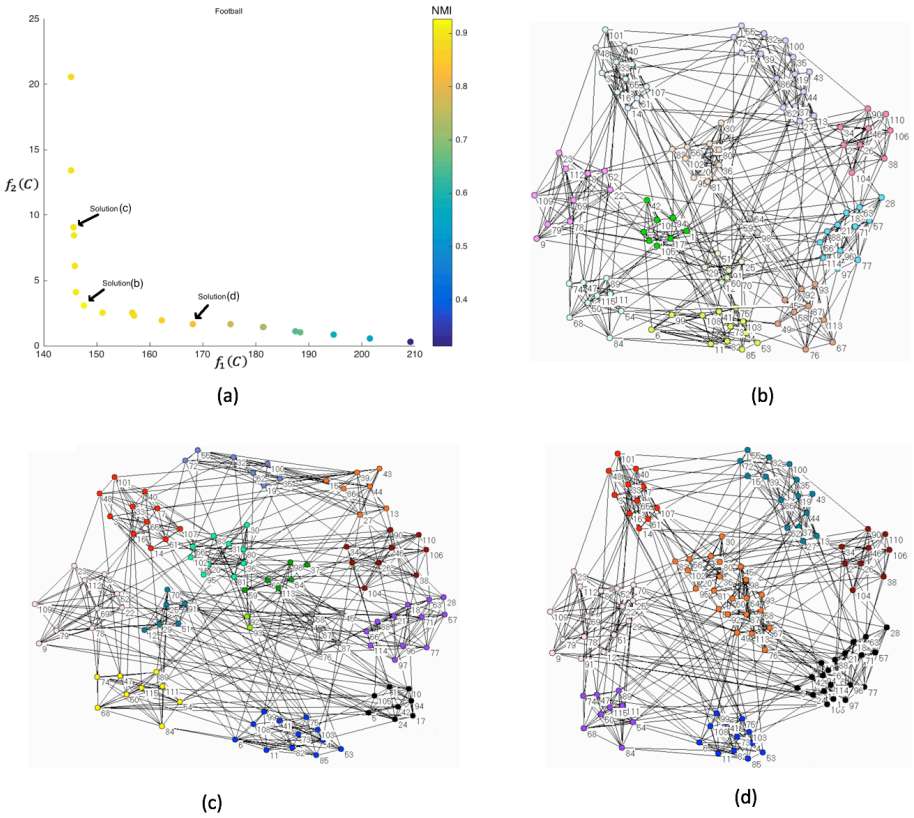


Figure 3.11 Community detection results on the Football network by MOEA-CD model. (a) Pareto front of one run with $NNC$ method. (b) Detected community structure which is corresponding to solution b at $NMI = 0.926879$. (c) Detected community structure which is corresponding to solution c at $NMI = 0.8940$, the network is divided into thirteen communities. (d) Detected community structure which is corresponding to solution d at $NMI = 0.8273$, the network is divided into eight communities. Colours indicate the community that a node belongs to.

models (MOEA-CD, MOCD and MODPSO) to reveal community structures.

Finally, for the Krebs' network, based on what is recorded in Table 3.4, our model classifies most nodes correctly and produces the best solution with $NMI_{max} = 0.6788$ while from the perspective of $NMI_{av}$, MOCD generates the best solution with $NMI_{av} = 0.6087$. As seen here, the results have shown the positive effect of the $NNC$ procedure with all models where this heuristic strategy overcomes the sensitivity of these models to local optima.

All the last three networks (Football 2000, Football 2001 and Krebs' networks) have weak nodes (the number of external connections being more than the number of internal connections), so the communities detected by the algorithm with the heuristic strategy could be better than the "true" partitions. For example, the

correct partition for the Football 2000 network has 15 weak nodes. However, the detected community structures by our algorithm have 10 weak nodes. This is because our model together with the $NNC$ strategy has assigned these nodes to what may be considered to be more meaningful communities. The correct partition is more likely to has communities that shared specific property and nodes within the community have internal connections more than external. In the same manner for the Football 2001 and Krebs' networks which have 9 and 15 weak nodes respectively.

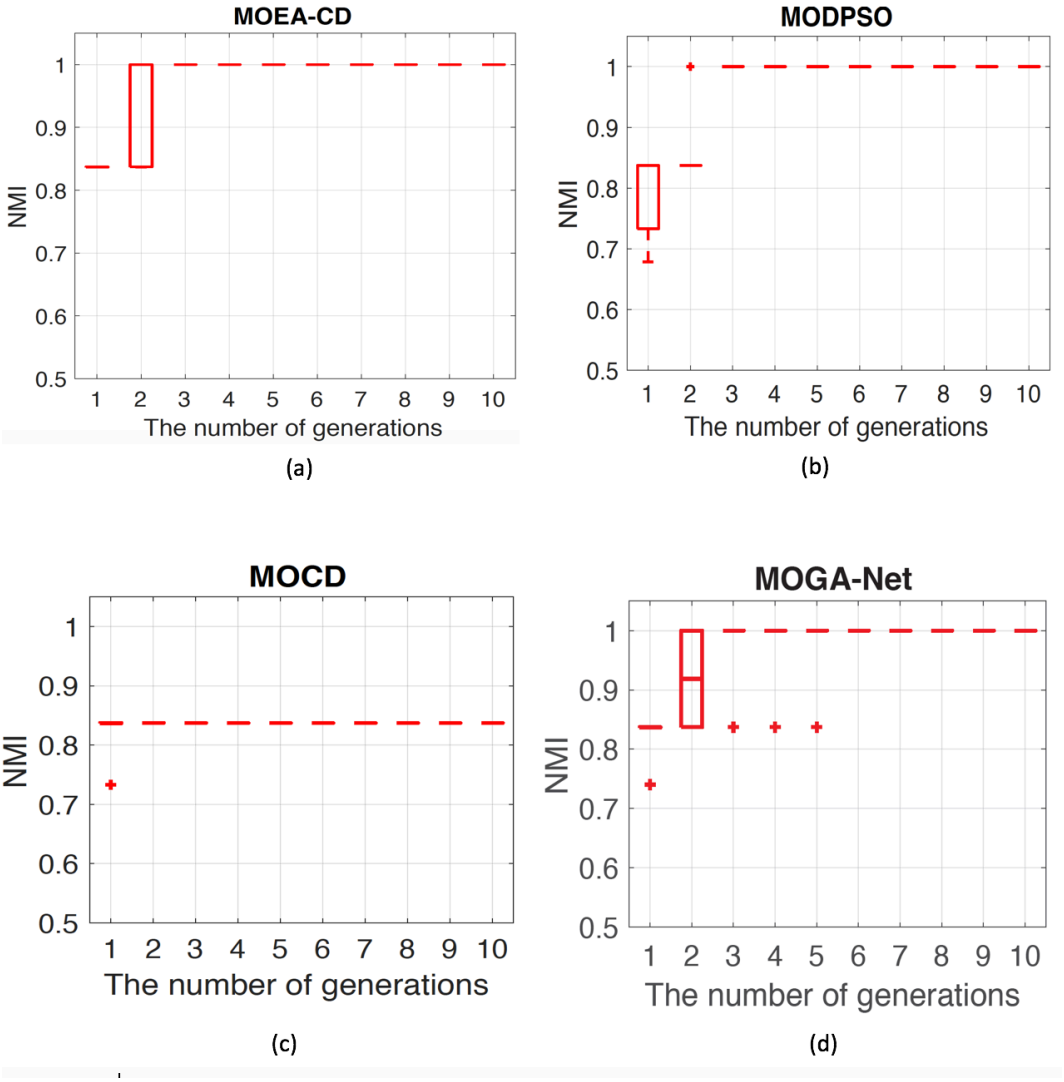


Figure 3.12 Box plots of the maximum $NMI_max$ between the detected partitions and the true partition versus generation on the Karate network. The box plots show the distribution of maximum $NMI$ over 20 runs for each of the four models: (a) Our proposed model; (b) MODPSO; (c) MOCD; (d) MOGA-Net.

Despite all models finding the correct partition, except MOCD which is trapped at a local optima at $NMI_{av} = 0.8372$, our model is the fastest to reach the optimal solution, as illustrated in Figures 3.12 and 3.13. As shown in Figure 3.12, our

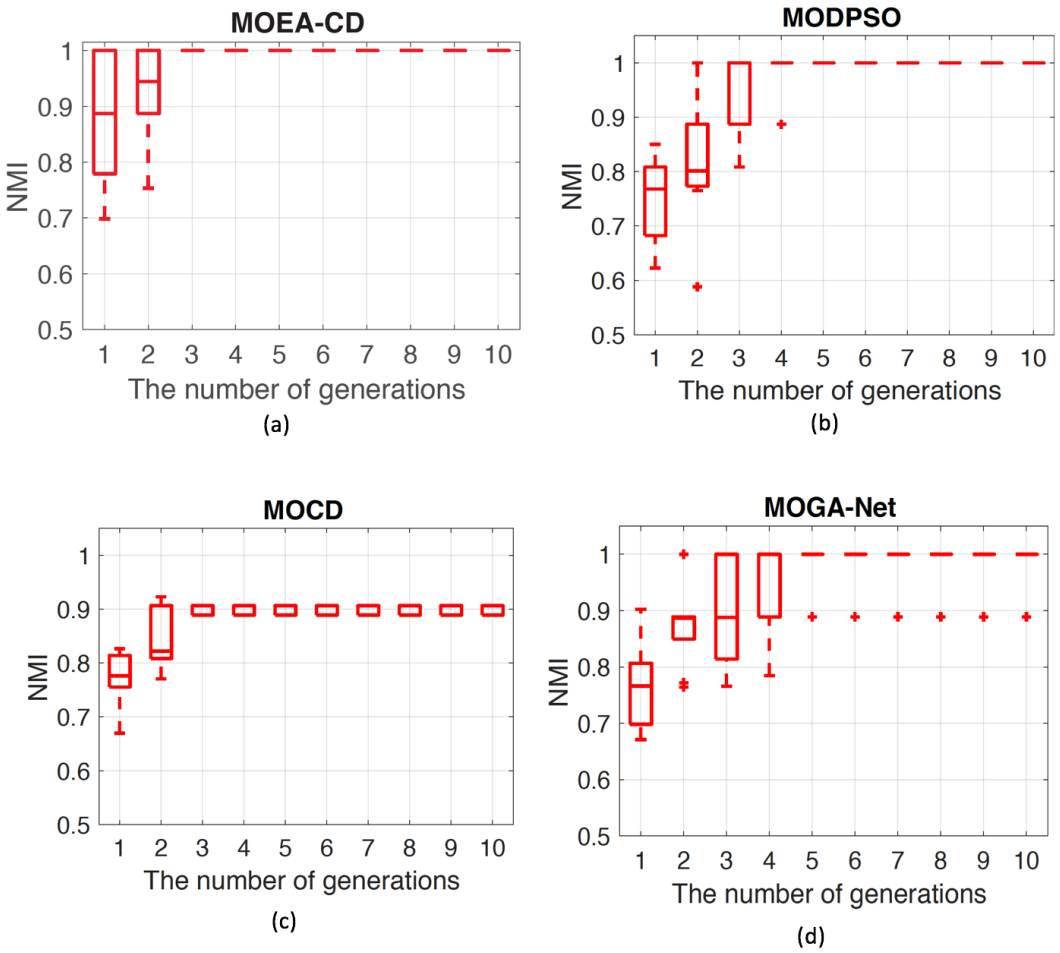


Figure 3.13 Box plots of the maximum $NMI$ between the detected partitions and the true partition versus generation on the Dolphin network. The box plots show the distribution of maximum $NMI$ over 20 runs for each of the four models: (a) Our proposed model; (b) MODPSO; (c) MOCD; (d) MOGA-Net.

model can detect the correct partition in the second generation in several runs while MODPSO for the same generation can detect correct partition in only one run, the most runs stuck in local optima at $NMI_{max} = 0.8372$. MOGA-Net can identify the correct partition in the second generation, but it is still stuck at $NMI_{max} = 0.8372$ until the fifth generation. MOCD cannot find the true partition in the first ten generations. In Figure 3.13, the results show also our model is faster one to find the true partition.

Table 3.5 shows the computational time in seconds for the four algorithms (MOCD, MOGA-Net, MOPSO and MOEA-CD) where these algorithms are different in only the objective functions. Although there are fluctuations in the running times among these algorithms, MOGA-Net has the longest running time due to the computation

Table 3.5 The average computational time in seconds over twenty runs of four algorithms (MOCD, MOGA-Net, MOPSO and MOEA-CD) per generation on real-world networks.

| Networks | MOCD | MOGA-Net | MOPSO | MOEA-CD |
|---|---|---|---|---|
| Karate | 0.14 | 0.2 | 0.1933 | 0.1567 |
| Dolphin | 0.4133 | 0.53 | 0.3967 | 0.3567 |
| Football2000 | 1.0433 | 1.37 | 1.1533 | 1.2333 |
| Football2001 | 1.1433 | 1.3467 | 1.1367 | 1.1167 |
| Krebs' books | 0.8333 | 0.9133 | 0.7200 | 0.7333 |

time for the Community Score (see Equation 2.21) which needs to calculate the internal connections for each node within the community and the internal connections for each community in the network. As we can see, the running time is longer when the size of the network is increased.

### 3.4.3 Real-world networks with unknown ground-truth partitions

In our final set of experiments, we investigate the performance of four models on real-world networks whose the ground-truth partitions are unknown. Table 3.6 compares all the models across real networks without the $NNC$ heuristic based on the statistical of the average maximum modularity $Q$ over twenty runs. $Q$ is used as the evaluation criteria rather than $NMI$ because the ground-truth partitions for these networks are unknown.

Firstly, we evaluate the four models on the SFI network [Girvan and Newman, 2002]. This is the network of collaborations of scientists at Santa Fe Institute in Santa Fe, New Mexico, USA during the calendar year 1999-2000. This network consists of 118 scientists who are represented by vertices. An edge exists between any two scientists if there is a collaboration between them due to publishing a paper together. There are 200 edges in this network.

Figure 3.14a shows the Pareto front that is produced by our algorithm. This is a set of nondominated solutions where each one represents a network partition and we choose the partition that has maximum modularity to represent the SFI network. Figure 3.14b illustrates the network partition into seven main communities.

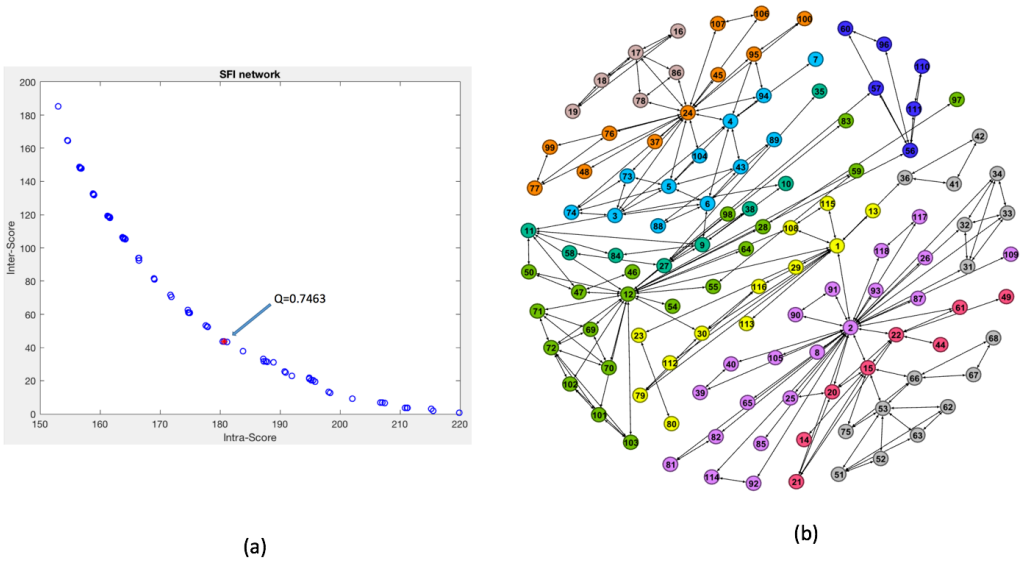


Figure 3.14 Community structure by MOEA-CD on the SFI network. (a) The trade-off set between Intra-Score and Inter-Score. Each blue circle in the estimated Pareto front is a solution that represents a different network partition to the SFI network. The red star is the network partition that corresponds to the solution at $Q = 0.763$. (b) SFI network is partitioned into seven main communities. Colours indicate the community that a node belongs to.

Table 3.6 Experimental results for testing four models without $NNC$ strategy on three real-world networks whose the ground-truth partition is unknown.

| Networks | Criteria | MOCD | MOGA-Net | MODPSO | MOEA-CD |
|---|---|---|---|---|---|
| SFI | $Q_{av}$ | 0.7338 | 0.7272 | 0.7373 | **0.7385** |
| Jazz | $Q_{av}$ | 0.2756 | 0.2387 | 0.3032 | **0.3178** |
| Netscience | $Q_{av}$ | 0.8956 | 0.8797 | 0.9054 | **0.9211** |

Secondly, the Jazz musician network consists of 198 bands which obtained from the Red Hot Jazz Archive digital database. These bands performed between 1912 and 1940 [Gleiser and Danon, 2003]. Finally the Netscience coauthors, this network contains authors working on network theory and experiments. It is interesting to note in Table 3.6 and Table 3.7 that our model can produce useful partitions without and with the $NNC$ procedure based on $Q_{av}$ for all these networks.

## 3.5 Summary

In this chapter, we have presented an evaluation methodology to assess the performance of objective functions that have been used for community detection on real-world networks based on their accuracy. The proposed method is based on a

Table 3.7 Experimental results for testing four models with $NNC$ strategy on three real-world networks whose the ground-truth partition is unknown.

| Networks | Criteria | MOCD | MOGA-Net | MODPSO | MOEA-CD |
|---|---|---|---|---|---|
| SFI | $Q_{av}$ | 0.7447 | 0.7394 | 0.7452 | **0.7463** |
| Jazz | $Q_{av}$ | 0.4141 | 0.4102 | 0.4313 | **0.4374** |
| Netscience | $Q_{av}$ | 0.9410 | 0.9281 | 0.9400 | **0.9427** |

random migration strategy to validate the quality of the existing and new scores. In other words, how well different definitions for the structure of communities augmented with the correct partition. In addition, although many algorithms have been proposed to solve community detection in static networks; this field need more investigation for discovering the accurate structure or analysing these communities. Therefore, we present the new Multi-Objective Evolutionary Algorithm for Community Detection (MOEA-CD). We optimise two conflicting objectives using MOEA/D where these objectives are derived from our investigation of node relationships in the networks. The first objective attempts to increase the number of connections inside the community (Intra-connections) and the second one to minimise the number of connections between different communities (inter-connections). In this case, a set of the best trade-off between these objectives is produced where each solution corresponding to different network partitions. These non-dominated solutions in the Pareto front are very important to investigate the analysing the community structures at the different level (variety of network partitions that are close to correct partition).

In this chapter, there is another improvement to the community detection algorithm that has been suggested. We proposed Neighbour Node Centrality as a heuristic mutation operator to speed up the convergence ability of the evolutionary algorithm.

We evaluate these partitions using $NMI$ and modularity measures. The experimental results show that our algorithm can accurately detect community structures compared with three state-of-the-art pairs of objectives on both synthetic and real-life networks. This algorithm opens up avenues for future work on weighted networks where the degree of correlation between nodes is considered. Therefore we need to

extend our objectives to work with this type of networks or others like signed or directed networks. Moreover another avenue that needs more investigation is how to choose the best solution among the set of estimated Pareto optimal solutions which are generated using an MOEA. The approximation set consists of the different structure of network partitions. Based on our experiments in this chapter, some of them are the true partition or close to the true partition. We used $Q$ to select the best solution. However, Figures 3.7 and 3.8 show that Q is quite a good method, but not perfect. Therefore, more investigations are needed in this area.

# Chapter 4

# Detecting Dynamic Communities Using Viterbi and Evolutionary Algorithms

Recently, the configuration of social networks is changed rapidly (like Facebook and Twitter) where communities are recognised by a sequence of evolutionary events (these events such as addition deletion, merge or split). The process of discovering the dynamics of these networks is challenging because this process needs to simultaneously identify community structures and their evolution over time. Therefore, a number of researchers have been motivated to analyse dynamic networks and proposed algorithms to find community structure in them. Perhaps the earliest works in this area are proposed by Hopcroft et al. [2004], who in 2004 introduced the first algorithm to detect dynamic communities in the NEC CiteSeer database. The agglomerative clustering method was used to find natural communities in each snapshot; after which similar communities at different times were grouped together.

One approach to analysing evolving communities is to detect communities at each timestep independently of other communities and then link the detected communi-

ties using a measure of their similarity [Leskovec et al., 2005; Kumar et al., 2005]. A weakness of these techniques, however, is that noise in the observed networks may yield quite dissimilar community structures which can be difficult to link [Lin et al., 2009; Kim and Han, 2009].

An alternative general approach, proposed by Chakrabarti et al. [2006] is to couple the detection of the communities at a particular timestep with the detected communities at the previous timestep. As briefly discussed in chapter 2, Chakrabarti et al. suggest two measures: the "Snapshot Cost" ($SC$), which measures the quality of the community structure and the "Temporal Cost" ($TC$), which penalises community structures which are dissimilar to the community structure at the previous timestep. For each timestep they, therefore, minimise a cost.

$$Cost = \alpha \times SC + (1 - \alpha) \times TC \tag{4.1}$$

where $\alpha$ controls the balance between detecting community structures that fit the observed network well and structures that are similar to those detected at the previous timestep. The range of $\alpha$ is between 0 and 1.

Rather than minimising a single, weighted cost, Folino and Pizzuti proposed a dynamic optimisation model by using a multi-objective evolutionary algorithm to find solutions which trade-off quality of the detected community structure (Snapshot Cost) at current timestep with the similarity to communities at the previous timestep (Temporal Cost) [Folino and Pizzuti, 2010]. They used Community Score as the first objective to maximise the quality of community structure at the current time step, while the second objective was the Normalised Mutual Information ($NMI$) [Danon et al., 2005] to measure the similarity between the community structure at the current timestep and the structure selected at the previous timestep. The $NMI$ measures the difference between the structures of communities over consecutive time steps and thus penalises dramatic shifts between successive time steps. Their algorithm, named DYNMOGA, employed the well-known multi-objective op-

timisation algorithm NSGA-II [Deb et al., 2002]. This algorithm was the first study to use multi-objective evolutionary algorithms to analyse the evolution of communities over time and their results outperformed previous studies such as [Lin et al., 2009; Kim and Han, 2009]. Subsequently, Folino and Pizzuti have investigated using other measures of the snapshot quality, including modularity, Community Score, CONductance and Normalised Cut [Folino and Pizzuti, 2014]. In a similar work [Ma et al., 2014] used modularity and $NMI$ as quality and temporal smoothing objectives, although with a different multi-objective evolutionary algorithm (MOEA/D) [Zhang and Li, 2007].

All the existing methods for analysing community evolution have used only one objective to evaluate the snapshot quality such as Modularity, Community Score, CONductance, Normalized Cut, etc [Folino and Pizzuti, 2010, 2014; Ma et al., 2014; Zhou et al., 2015] while community detection is often beneficially treated as a multi-objective problem due to networks have multiple structure properties [Shi et al., 2012; Pizzuti, 2012; Gong et al., 2014; Wu and Pan, 2015]. That motivates us to employ two objectives to evaluate the snapshot quality at each snapshot.

In this chapter, we view the communities themselves as evolving according to a Markov model, with observations at each time step governed by the latent state of the communities. However, the straightforward application of filtering and smoothing algorithms based on Hidden Markov Models (HMMs) is hampered by the vast number of possible states—partitions of nodes into communities—for any real network. To combat this, we use a multi-objective evolutionary algorithm to locate a small number of probable states at each time step. Within the space of these probable states, we then use the Viterbi algorithm [Rabiner, 1989] which is a dynamic programming algorithm to find the most probable sequence of states, that is the most probable sequence of communities.

In order to find probable candidate states, we simultaneously optimise two objectives as functions of the community structure. Communities are characterised by dense

connections within each community and sparse connections between them. The first objective (the Intra-Score (Equation 3.1 )) therefore quantifies the density of links within communities, while the second objective (the Inter-Score (Equation 3.3)) measures inter-community sparsity. As described in chapter 3, we adopt our algorithm MOEA-CD to locate an approximation to the Pareto front, the optimal trade-off set between the two objectives. As shown in chapter 3, this algorithm is able to locate a wide range of network partitions that are close to the true partition. We generate approximations to the Pareto-optimal solutions at each time step and then use the Viterbi algorithm to find the most likely sequence of communities from within these candidate sets. This sequence of communities has the minimum temporal transition cost between the different Pareto sets. The structures of these communities represent the best network partitions that could be the true partitions or very close to the true partitions.

This study is different from existing algorithms in two aspects. First, detecting communities at each time step separately by optimising two conflicting objectives then the most likely partitions are found over different time steps. This idea comes from our investigation of community detection in a static network when network partitions are evaluated by using multi-objectives to produce a more accurate structure than the single objective optimisation. The other aspect is that this algorithm can produce the most likely sequence of partitions among the available Pareto optimal solutions (states) by using the Viterbi algorithm for the dynamic networks when the true partition for a given network is known or unknown. The Viterbi algorithm is a common method to produce the most likely sequence of states for different purposes.

The main contributions of this chapter lie in formulating the detection of dynamic communities as an HMM to capture the evolution of these communities, and the use of an MOEA to produce the candidate states at each time step.

This chapter is organised as follows. We first describe our formulation of dynamic community detection as a hidden state problem, we describe the multi-objective

evolutionary algorithm used to locate candidate states, after which the algorithm is demonstrated on synthetic and real dynamic networks.

## 4.1 Dynamic Community Detection with HMMs

We now a formulate the problem of detecting dynamic communities in a hidden Markov model framework. We model a dynamic network as a sequence of graphs $G = (G^1, G^2, \ldots, G^T)$ observed over $T$ discrete time steps. For simplicity, each graph is considered as undirected and unweighted. Each observed graph is $G^t = (V, E_t)$ where $V$ represents set of nodes in the network, which for simplicity we regard as fixed in number (although perhaps not all observed and we assign the nodes that are not observed to a community zero). Let $V(G) = \{v_1, v_2, \ldots, v_N\}$ with $N = |V|$ and $E(G^t)$ represents a set of links between nodes at time step $t$ in $G^t$. We denote by $L_t$ the number of edges in the graph at time $t$; $L_t = |E(G^t)|$.

Let $G^t$ be represented as an $N \times N$ adjacency matrix $A^t$ so that $A^t_{ij} = 1$ if there is a link between $v_i$ and $v_j$, while $A^t_{ij} = 0$ otherwise. $\sum_{j=1}^{N} A^t_{ij} = 0$ for the unobserved node $i$ at any $t$ and we supposed the community of this node is 0. At each time step $t$ we model the graph $G^t$ as partitioned into $K_t$ communities $\{C^t_i\}_{i=1}^{K_t}$ so that each node belongs to exactly one community. We regard the community structure as a latent variable whose value is unobserved. Let the set of all partitions be $\Omega$. Then, clearly the number of possible partitions is $2^N$. However, rather than consider all these hidden states we restrict the model to consider a smaller number $M \ll 2^N$ of more likely configurations. We denote by $\mathbf{c}_t$ the $M$-dimensional vector specifying which of the $M$ states/partitions the graph is in at time $t$.

Community membership itself is unobserved. Instead, observations comprise the links (edges) between some of the nodes, so that the entire observation at time $t$ is captured by the adjacency matrix $A^t$ of the graph $G^t$. The emission probability of observing a particular adjacency matrix models how well a particular community

structure $\mathbf{c}_t$ fits the observed adjacency matrix $A^t$. For example, the modularity $Q(G^t, \{C_i^t\}_{i=1}^{K_t})$ is a popular measure for evaluating community structure [Newman and Girvan, 2004]:

$$Q(^t, \mathbf{c}_t) = \sum_{t=1}^{K_t} \left[ \frac{\underline{D}(C_i^t)}{2L_t} - \left( \frac{D(C_i^t)}{2L_t} \right)^2 \right] \tag{4.2}$$

where we regard $\mathbf{c}_t$ as specifying the partition $\{C_i^t\}_{i=1}^{K_t}$ and the degree $D(C)$ and internal degree $\underline{D}(C)$ of a community $C$ are defined in chapter 2(see Equations(2.2 and 2.4) respectively):

The modularity may be shown to be the summed differences between the fraction of links within a community minus the expected fraction of links within the community if the graph were rearranged at random but preserving the degree distribution [Newman and Girvan, 2004]. Partitions of the network that have high values of modularity, therefore, have dense connections within the community and sparse links with the others. The modularity may be used to define the probability of observing the network $A^t$ given a community structure $\mathbf{c}_t$ as follows:

$$p(A^t \,|\, \mathbf{c}_t) \propto Q(A^t, \mathbf{c}_t). \tag{4.3}$$

Thus adjacency matrices that conform well to a particular latent community structure are regarded as probable. Other measures of the community structure might be used in place of $Q$.

Temporal smoothness is incorporated into hidden Markov models via the transition probability. Our model for the transition probability between states encodes the belief that transitions between similar states are more likely than those between dissimilar states; that is, the network tends to evolve slowly, making small transitions. We model the probability of a transition from $\mathbf{c}_{t-1}$ to $\mathbf{c}_t$ as:

$$p(\mathbf{c}_t \,|\, \mathbf{c}_{t-1}) \propto \mathrm{NMI}(\mathbf{c}_t, \mathbf{c}_{t-1}) \tag{4.4}$$

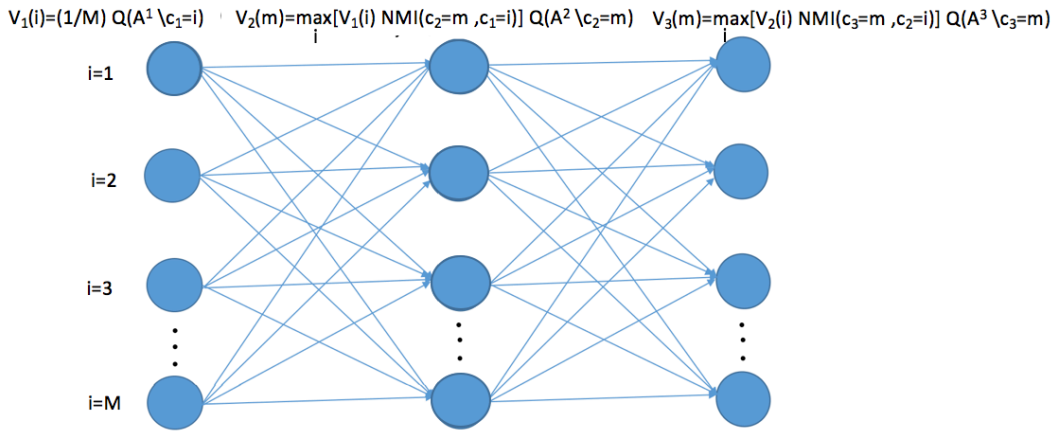


Figure 4.1 An example of the Viterbi algorithm captures the evolution of dynamic communities over three time steps.

where $\text{NMI}(\mathbf{c}_t, \mathbf{c}_{t-1})$ is the Normalised Mutual Information [Danon et al., 2005] between the partitions specified by $\mathbf{c}_{t-1}$ and $\mathbf{c}_t$. The $NMI$ is commonly used to compare the similarity of cluster or community configurations and has been used by other authors to penalise abrupt transitions between community structures [Folino and Pizzuti, 2010, 2014; Ma et al., 2014].

As we mentioned earlier, we consider the number of nodes as constant despite the fact that some nodes may be not observed either hide or birth at all time steps. This number is determined by the maximum observed nodes over different time steps. In this case, we can calculate $NMI(\mathbf{c}_t, \mathbf{c}_{t-1})$ even when the number of visible nodes for each partition changes over time [Folino and Pizzuti, 2014; Ma et al., 2014]. With the probability of transitions between states (partitions) and the the probability of observing a graph given the latent partition defined by (4.4) and (4.3), the well-known Viterbi algorithm may be used to find the most likely sequence of states–the Viterbi path–to have given rise to the observations [Rabiner, 1989]. The Viterbi algorithm is initialised with the probabilities of the initial state $p(\mathbf{c}_0 = m) = \pi_m$ for $m = 1, \ldots, M$. Then define $v_t(m)$ be the value of the $m$th state at time $t$, which is proportional to the probability that the most probable path ends at time $t$ in state

**Algorithm 4.1** Viterbi algorithm for capturing the evolution of dynamic communities.

Inputs
1 : $A$ : Adjacency matrix.
2 : $N$ : Number of nodes in the network
Steps
1 : $Initialisation : v_1(i) = \frac{1}{M} Q(A^1 \,|\, \mathbf{c}_1 = i)$
2 : $Path(i) = 0$
3 : $Recursion : v_t(m) = \max_i v_{t-1}(i) Q(A^t \,|\, \mathbf{c}_t = m)\, \mathrm{NMI}(\mathbf{c}_t = m, \mathbf{c}_{t-1} = i)$
4 : $Path(m) = arg \max_i v_{t-1}(i) Q(A^t \,|\, \mathbf{c}_t = m)\, \mathrm{NMI}(\mathbf{c}_t = m, \mathbf{c}_{t-1} = i)$
5 : $Termination : \mathbf{c}_T^* = arg \max_i v_T(i)$
6 : $Backtracking : \mathbf{c}_t^* = path_{t+1}(\mathbf{c}_{t+1}^*), t = T-1, T-2, ..., 1$

$m$. Then $v_t(m)$ is recursively updated as:

$$v_t(m) = \max_i v_{t-1}(i) p(A^t \,|\, \mathbf{c}_t = m) p(\mathbf{c}_t = m \,|\, \mathbf{c}_{t-1} = i) \tag{4.5}$$

$$= \max_i v_{t-1}(i) Q(A^t \,|\, \mathbf{c}_t = m)\, \mathrm{NMI}(\mathbf{c}_t = m, \mathbf{c}_{t-1} = i) \tag{4.6}$$

Once the end of the sequence is reached, the finishing state of the most probable sequence is identified and back pointers (constructed during the forward sweep) used to recover the most probable sequence of states leading to it. We denote this Viterbi path/sequence of states by $\{\mathbf{c}_t^*\}_{t=1}^T$. Figure 4.1 shows an example of the calculation of Viterbi algorithm over three time steps. The Viterbi algorithm is shown in Algorithm 4.1.

The computational complexity of the Viterbi algorithm is proportional to the number of observations and the square of the number of states. In practice, the overall computational time is dominated by the time to discover candidate partitions for all the timesteps.

### 4.1.1 Multi-Objective Evolutionary Algorithm

We use the same algorithm for community detection in static networks (MOEA-CD) by optimising two objectives that have been used in chapter 3, one quantifying the density of internal connections within communities and the other quantifying the

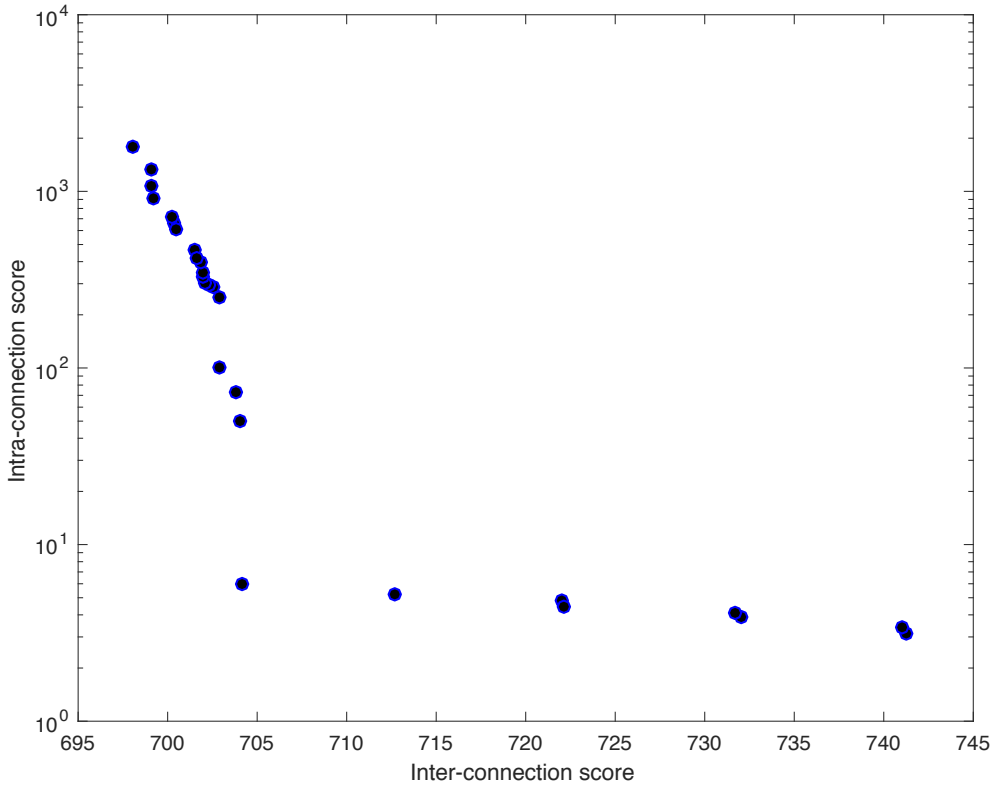


Figure 4.2 The trade-off set between the Inter-Score and Intra-Score for a single snapshot ($t = 5$) for *Var-Net*, $z = 5$ data which will be described in section 4.2.

sparsity of connections between communities. This algorithm generates solutions which represent a variety of partitions of the network. These solutions trade-off the two objectives. Full details of the algorithm, genetic representation and heuristics to improve the convergence rate are given in chapter 3.

As an illustration, Figure 4.2 shows the approximation to the Pareto optimal set resulting from 300 iterations of the MOEA on a single snapshot ($t = 5$) for the *Var-Net*, $z = 5$ dataset (see below) for optimising the inter-score and the Intra-Score objectives. This set consists of 27 mutually non-dominating solutions. We use the set of approximations to the Pareto optimal solutions located like this as the basis for the Hidden Markov Model.

## 4.2 Results

In this section, we present and discuss the results which show the efficacy of our proposed algorithm and compare the results obtained by our algorithm with the algorithm of Lin et al. [2009], Kim and Han [2009] and Folino and Pizzuti [2014] on synthetic networks for which the true partitions are known. We first illustrate these algorithms on eight synthetic networks drawn from the literature for which the correct partitions are known. Subsequently, we apply our algorithm to two real datasets, one the well-known Paraiso cell-phone network [Grinstein et al., 2008] and the other a new data set concerning tweets between British Members of Parliament during the weeks preceding the Brexit referendum [Weaver et al., 2018].

In all the results shown the MOEA was run 5 times for 300 iterations on each snapshot with a neighbourhood size of 5 and the union of the results of each run used as the set of candidate hidden states for that snapshot $t$.

### 4.2.1 Synthetic Datasets

#### Kim and Han datasets.

The first synthetic datasets that we examine were proposed by Kim and Han [2009]. Each is formed of 10 consecutive snapshots of the graph as it evolves: $G = (G_1, G_2, \ldots, G_{10})$. To generate dynamically evolving networks, some of the nodes leave their home communities in $G_{t-1}$ and are assigned randomly to other communities in $G_t$. The parameter $z$ determines the number of inter-connections made by a node: increasing $z$ leads to noisier network structures. In these experiments, we use data with $z = 3$ and $z = 5$.

The *Fix-Net* dataset comprises a fixed number of communities, while the number of communities varies with time in the *Var-Net* data. In *Fix-Net* there are 128 nodes,

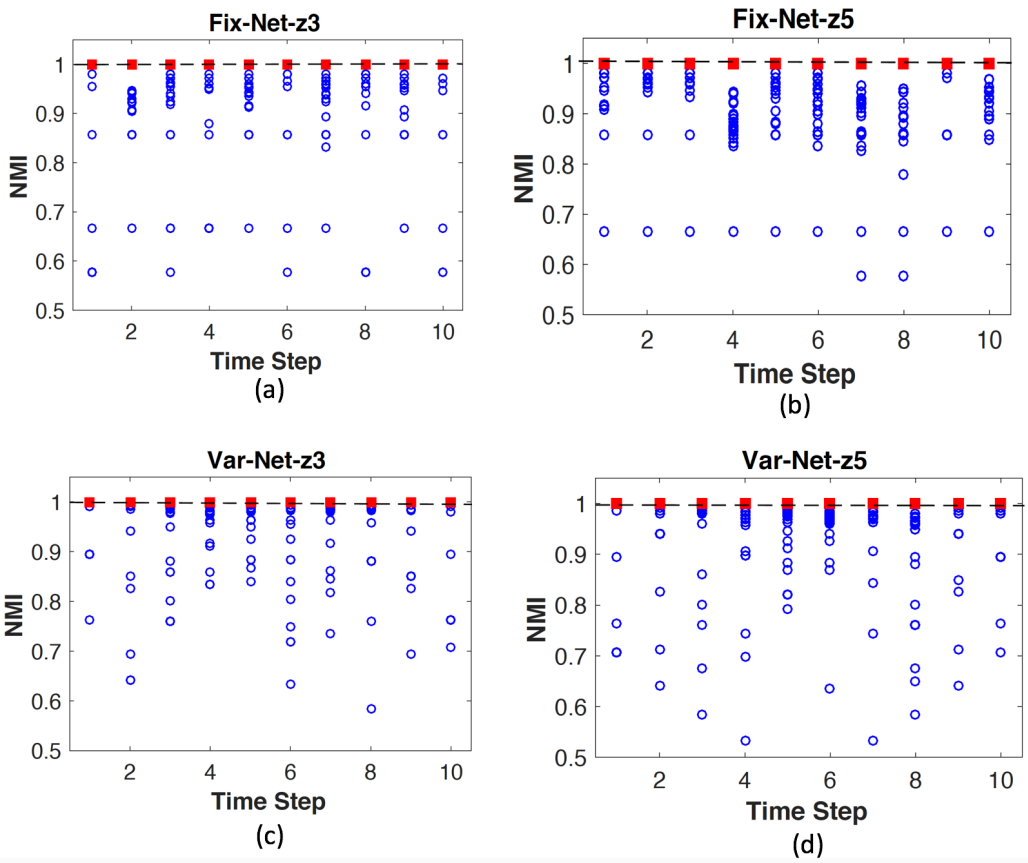


Figure 4.3 Kim and Han [2009] synthetic networks. $NMI$ between candidate partitions located by the MOEA and the true partition at each timestep (blue circles). $NMI$ between the true partitions and the Viterbi optimal path of partitions $\mathbf{c}_t^\star$ are shown as red squares.

which form 4 equally-sized communities of 32 nodes each. The average degree of each node is 16. At time step $G_{t-1}$, three nodes are selected randomly from each community and join randomly to three other communities in the time step $G_t$.

The number of communities in the *Var-Net* data varies during the evolution of the network. Initially, it has 256 nodes partitioned into 4 communities of 64 nodes each. The average degree of each node is half the size of its community. During the succeeding timesteps ($2 \leq t \leq 10$) 16 nodes are deleted at random from the network and 16 new nodes are added randomly. Furthermore, during the first half of the evolution ($1 \leq t \leq 5$) eight nodes are chosen at random from each community in $G_{t-1}$ and combined to produce a new community in $G_t$; during subsequent timesteps, the nodes are returned to their initial communities. Thus the number of communities during the 10 timesteps is $4, 5, 6, 7, 8, 8, 7, 6, 5$. Direct visualisation of these networks at each timestep is not revealing and consumes a lot of space, so we, therefore,

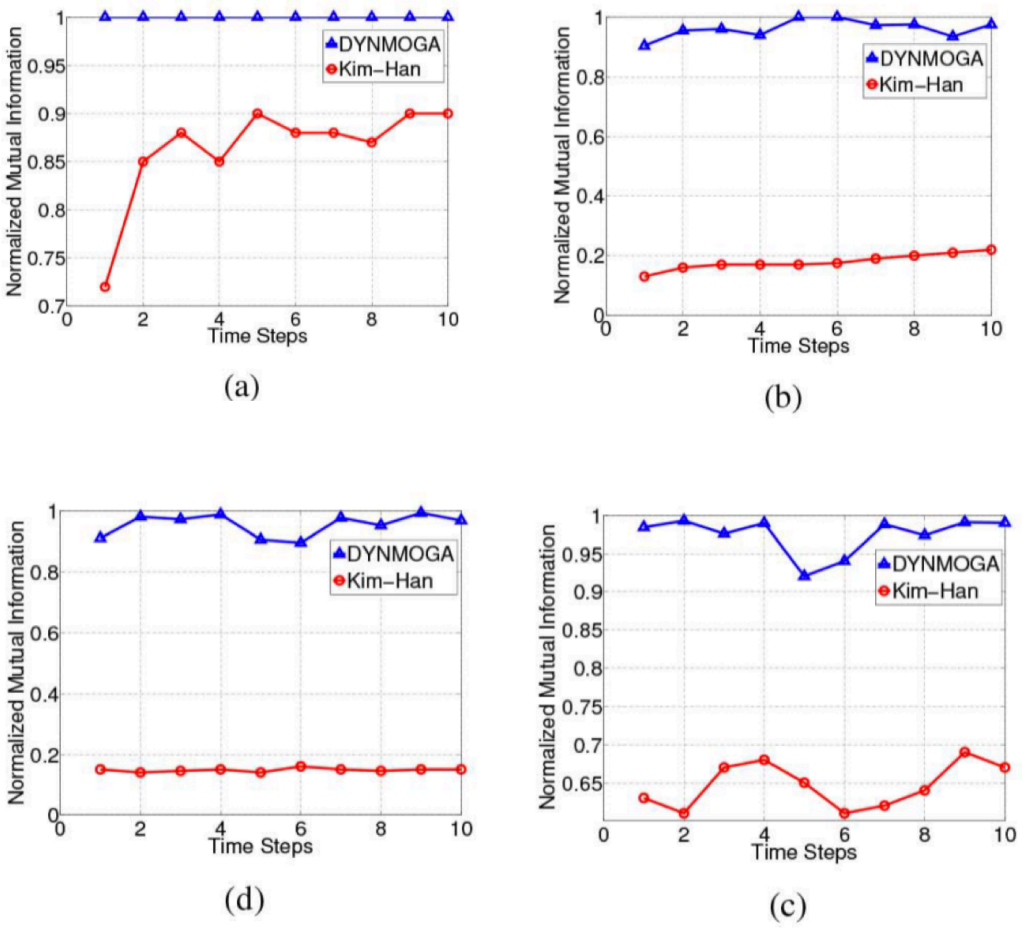


Figure 4.4 Kim and Han [2009] synthetic networks. $NMI$ between partitions detected by the DYNMOGA and Kim-Han and the true partition with blue and red lines respectively over 10 time steps. (a) Fix-Net-z3. (b) Fix-Net-z3. (c)Var-Net-z3. (d) Var-Net-z5. The figure was taken from Folino and Pizzuti [2014]

calculate at each timestep the normalised mutual information between the partition $\mathbf{c}_t^\star$ located by the Viterbi algorithm and the true partition, which we denote by $\mathcal{C}_t$, namely: NMI($\mathbf{c}_t^\star, \mathcal{C}_t$). Figure 4.3 shows, for each timestep, the Normalised Mutual Information between $\mathcal{C}_t$ and each of the members of the Pareto set comprising the candidate hidden states (blue circles). In addition, mutual information for $\mathbf{c}_t^\star$ is indicated by a red square. As the figure shows, the MOEA algorithm has located a range of candidate solutions, some of them close to the true partition and some of them distant. However, the hidden Markov model formulation and the Viterbi algorithm identifies the sequence of solutions that are closest to the true sequence of states. With one exception, the $NMI$ between the true partitions and Viterbi path partitions is 1, indicating that the correct sequence of partitions has been located. The single exception is the first timestep for *Fix-Net* with $z = 3$ (Figure

4.3a) in which the Viterbi path identifies the candidate solution second closest to the true solution, even though the true partition has been located by the MOEA. We attribute this to the choice of uniform initial probabilities $\pi_m$ in the Viterbi algorithm (Equation 4.5) which biases the initial state away from the true partition.

Figure 4.4 shows the average value of $NMI$ obtained by the DYNMOGA [Folino and Pizzuti, 2014] and the Kim-Han algorithms [Kim and Han, 2009] on Kim and Han datasets. The results founded by the DYNMOGA outperform Kim-Han's algorithm specifically when $z = 3$. However, our results outperform both the DYNMOGA and Kim-Han as our algorithm located the true partitions on the four networks due to the collaboration of two objectives: $f_{Intra}$, $f_{Inter}$ to evaluate the snapshot quality. $f_{Intra}$, $f_{Inter}$ are optimised to generate the possible partitions at each time step and modularity measures how well a particular partition fits the observed adjacency matrix. As a result, we conclude that Intra-Score and Inter-scores provide better results if both of them are used to evaluate snapshot quality.

**Green et al. datasets.**

The second set of datasets for evaluating dynamic networks has been proposed by Greene et al. [2010]. They developed four benchmarks each with 1000 nodes evolving over 5-time steps. The benchmarks are described briefly as follows:

**Birth and death: BD-Net.** At each time step some nodes leave their original communities and are combined to create new communities. 10% of existing communities are dissolved, and 10% of new communities emerge. The number of communities at each time step is 33.

**Expansion and contraction: EC-Net.** At each time step 10% of randomly selected communities expand or contract by 25% of their size. Nodes are joining expanding communities or leaving shrinking communities are selected at random.

**Intermittent communities: H-Net.** All nodes in 10% of communities are not ob-

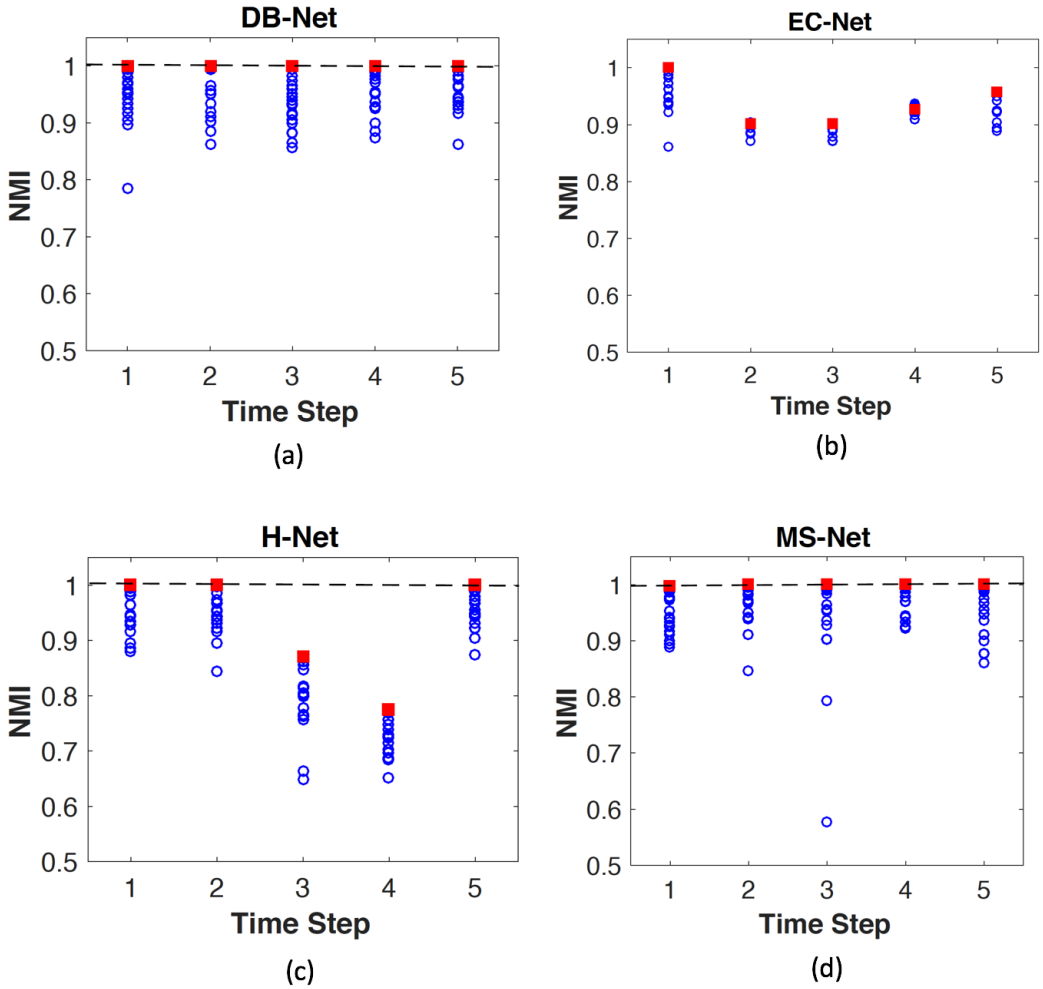


Figure 4.5 Greene et al. [2010] synthetic networks. $NMI$ between candidate partitions located by the MOEA and the true partition at each timestep (blue circles). $NMI$ between the true partitions and the Viterbi optimal path of partitions $\mathbf{c}_t^\star$ are shown as red squares.

served from the second time step onwards.

**Merging and splitting: MS-Net.** At each time step, 10% of randomly selected communities are split, and 10% of communities are merged.

Figure 4.5 shows the performance of the proposed algorithm. On the birth-death network, the MOEA has located partitions that include the true partition, and the Viterbi algorithm has correctly identified the true partition at all time steps. Likewise, partitions including the correct partition have been located and the correct partition identified by the Viterbi algorithm for the merge-split networks.

The evolution of the intermittent networks and the expansion and contraction networks produce many nodes that are "weak" in the sense that they have a high propor-

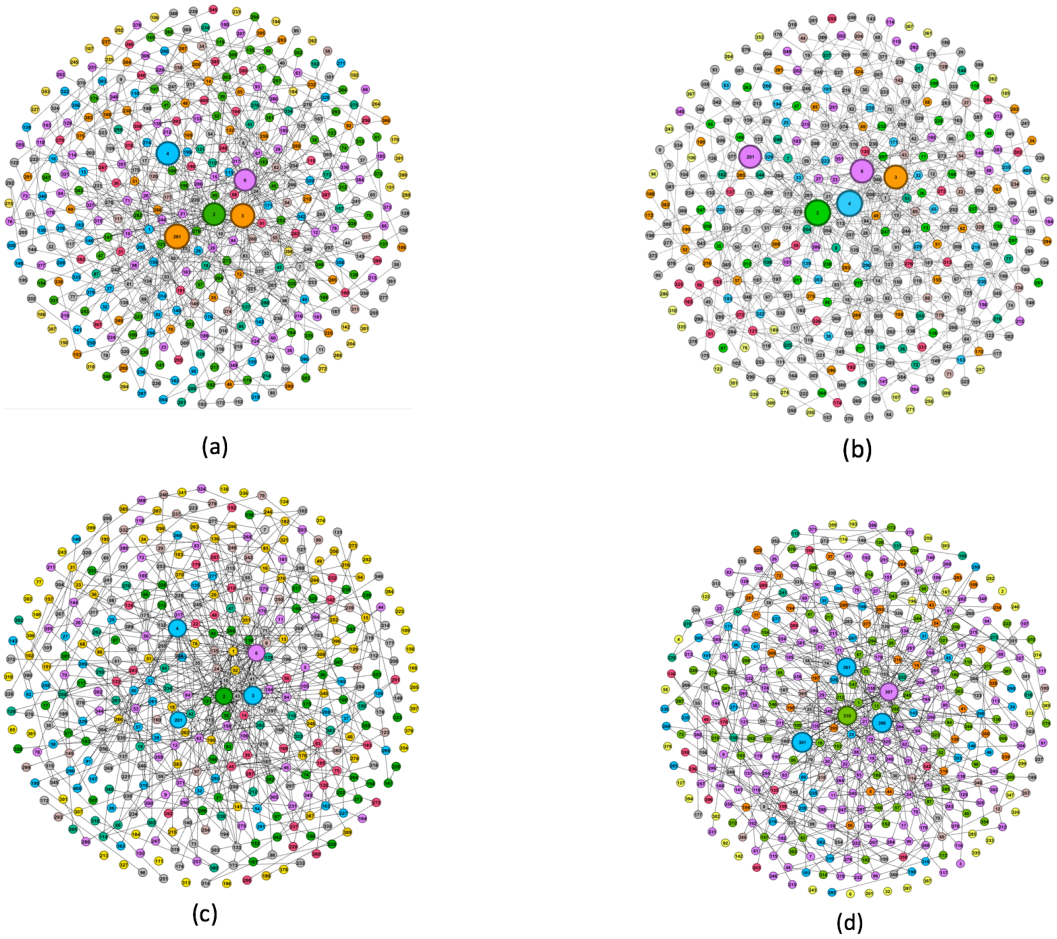


Figure 4.6 Cell phone calls networks. Community structures which are detected by our algorithm on the cell phone network. (a): Community structure on day one. The five important nodes in this partition (nodes 2(green), 3(orange), 4(blue), 6(purple) and 201(orange)) are assign to four communities. (b) partition is divided into four communities on day six, the five important nodes (nodes 2(green), 3(orange), 4(blue), 6(purple) and 201(orange)) are assigned to four communities. (c) Community structure on day seven. The important nodes are (2(green), 3(blue), 4(blue) and 6(purple)) are assigned to three communities. (d) Community structure on day eight. The important nodes 2, 3, 4, 6 and 201 changed their number to 310(green), 398(blue), 361(blue), 307(purple), 301(blue))

.

tion of connections to nodes outside their community rather than intra-community connections. In these cases, the MOEA has not always located a Pareto set which includes the correct partition. Nonetheless, we emphasise that the hidden Markov model formulation and the Viterbi algorithm have in each case identified the most similar partition to the true partition among the partitions (states) located by the evolutionary algorithm. It is possible that augmenting the set of candidate partitions with a richer set of partitions, such as those found during the evolutionary search, might find additional partitions closer to the true partition.

### 4.2.2 Real-life Datasets

Of course, the real networks are interesting to evaluate the proposed algorithm as these networks reflect different statistical properties of networks. Our algorithm is evaluated on two the following real-world networks:

**Cell phone call**: This network consists of 400 Paraiso cell-phones which introduced by IEEE Visual Analytics Science and Technology (VAST) 2008 Challenge [Grinstein et al., 2008]. It is ten days data sets based on the cell phone call of the Catalano/Vidro social communication in June 2006 in the Isla Del Sueno. Each node represents one cell phone, and an edge occurs between two nodes if there is a phone call between them. In this network, five persons that are considered as more active nodes than the others: Ferdinando Catalano (node 201) and his brother Estaban Catalano (node 6), David Vidro (node 2), and his two brothers Jorge and Juan represent in nodes 3 and 4 respectively. After day 7, these five members change their phone call numbers to 301, 307, 310, 398 and 361 until day 10.

Figure 4.6 shows the visualisation of network partition at time step 1, 6, 7 and 8 where these partitions are produced by evaluating our algorithm on cell phone benchmark. There are four important communities that describe in this figure at time step 1 where nodes (2, 4, 6) are assigned to three communities and nodes 3 and 201 are grouped in one community. These four important communities are evolving at time step 6, 7 and 8, as described in this figure. At day 8, these nodes (2, 3, 4, 6, 201) changed their number to 310, 398, 361, 307, 301 respectively.

#### MP Twitter network

Finally, we illustrate our algorithm by applying it to the evolving network of Twitter connections between UK Members of Parliament (MPs) in the 85 consecutive weeks from December 2014 to August 2016, the period including a general election on 7th

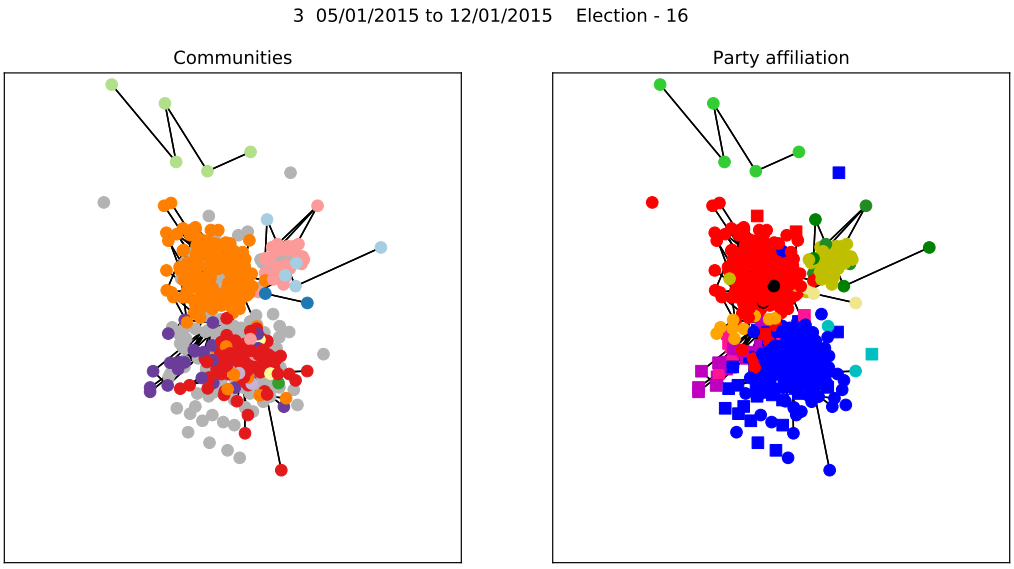


Figure 4.7 Communities and party affiliations for the MP twitter communities data. *Left:* Communities discovered by the evolutionary and Viterbi algorithms. Nodes representing MPs belonging to the same community are depicted in the same (arbitrary) colour; grey symbols indicate MPs who did not tweet that week. *Right:* Political party affiliation of the MPs: red: Labour; blue: Conservative; yellow Liberal Democrat; cyan: Scottish National Party; purple: United Kingdom Independence Party (UKIP); dark green: Plaid Cymru; black: speaker and independent.

May 2015 and the Brexit referendum held on 23rd June 2016 [Weaver et al., 2018]. This network consists of 648 nodes corresponding to the MPs and a link between nodes is made when one MP names another in at least one tweet that week. Of course, not all MPs tweet each week and 21 MPs did not use Twitter at all during the 85 weeks, so that the effective network consists of 626 nodes.

For much of the time, we find that the MPs may be divided into four main communities roughly corresponding to the political parties of the MPs, but with a number of (short-lived) smaller communities present. For example Figure 4.7 shows a visualisation of the communities for the week 05/01/2015 to 12/01/2015. Comparison of the left and righthand panels shows that the larger political parties for the main communities, but there are smaller communities, such as MPs in Sinn Fein depicted in light green (left panel) and dark green (right panel) near to the top of the figures. As Figure 4.8 illustrates we find in contrast to MPs of other parties, that the Conservative Party MPs sometimes tend to form much smaller, short-lived communities, rather than a single community.

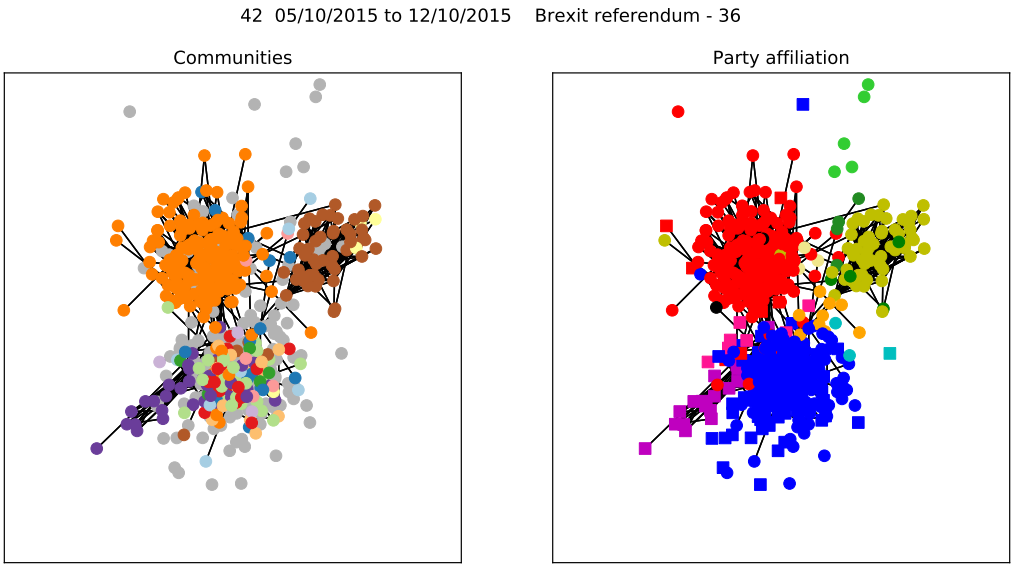


Figure 4.8 Communities and party affiliations for the MP twitter communities data illustrating the many smaller communities formed by Conservative Party MPs in contrast to the single larger communities representing other political parties. *Left:* Nodes representing MPs belonging to the same community are depicted in the same (arbitrary) colour; grey symbols indicate MPs who did not tweet that week. *Right:* Political party affiliation of the MPs: red: Labour; blue: Conservative; yellow: Liberal Democrat; cyan: Scottish National Party; purple: United Kingdom Independence Party (UKIP); dark green: Plaid Cymru; light green: Sinn Feinh; black: speaker and independent.

The visualisation was produced using a force-directed algorithm [Fruchterman and Reingold, 1991] with the spring constant for edges in the same community a factor of 5 larger than edges connecting nodes in different communities. The force-directed the algorithm to obtain the visualisation for each week's data was initialised using the node locations resulting from the previous week's data. Since the nodes comprising a community may change at each timestep, corresponding communities at successive time steps were identified using the Hungarian algorithm [Kuhn, 1955] with the similarity of communities being proportional to the number of MPs that were members of both communities divided by the total number of MPs in the two communities. Therefore, the Hungarian algorithm solves the assignment problem by finding the best matching between communities over different time steps. In other words, we find the weight of similarity in terms of a number of nodes that share between the community at time step $t$ and time step $t-1$. Then a community label at time step $t-1$ that has maximum weight is assigned to each node in the community at time step $t-1$. In this case, we keep nearly the same community label for each node over time. This is very important to track communities over time.

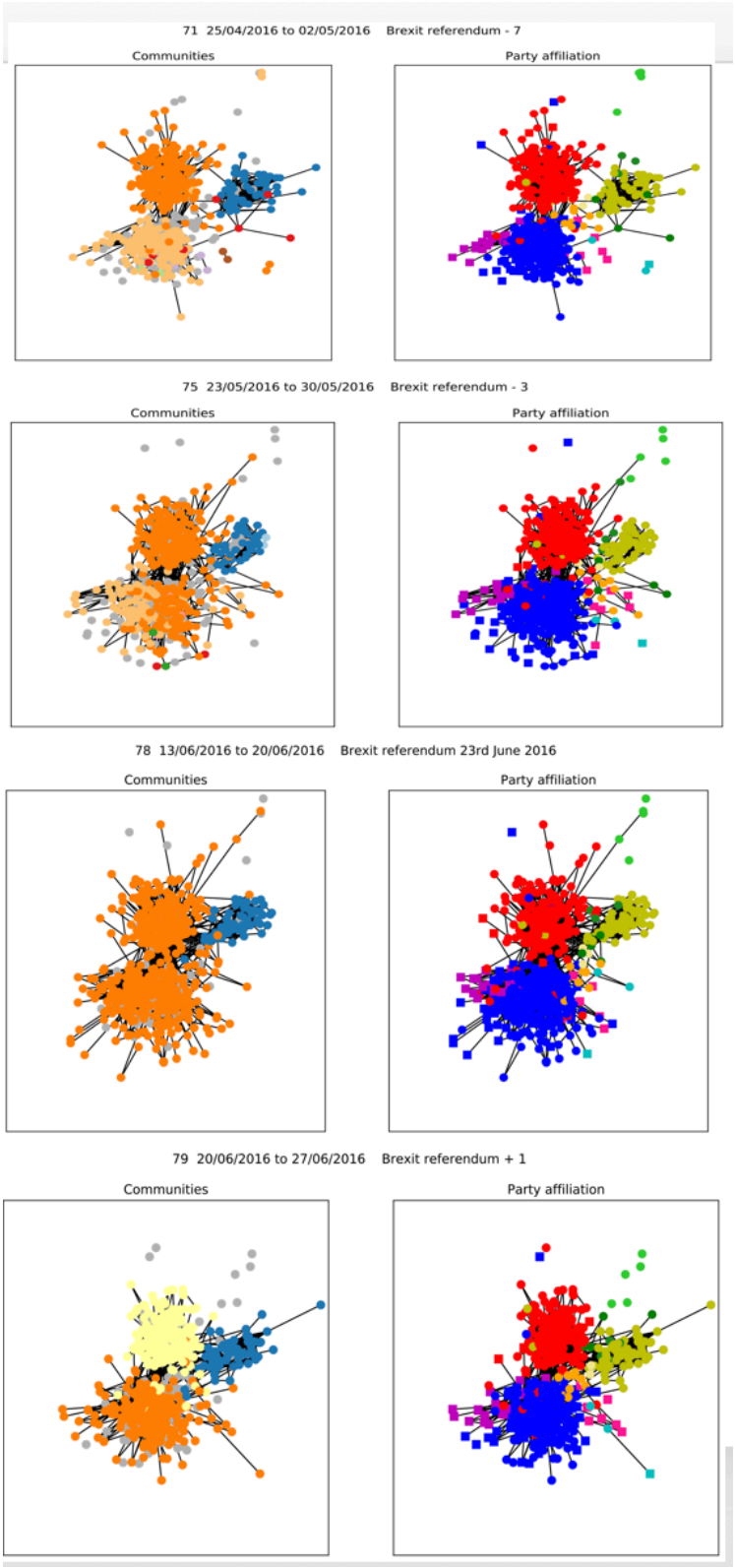


Figure 4.9 Weekly MP Twitter communities before and after the Brexit referendum. Successive panels show the community structure and party affiliations 7 and 3 weeks before the referendum, the week of the referendum and the week immediately following it. As the referendum approaches, two of the three main communities (Labour, Conservative plus UKIP, and Scottish National Party) merge to form a single community, which immediately after the referendum again splits apart along party lines.

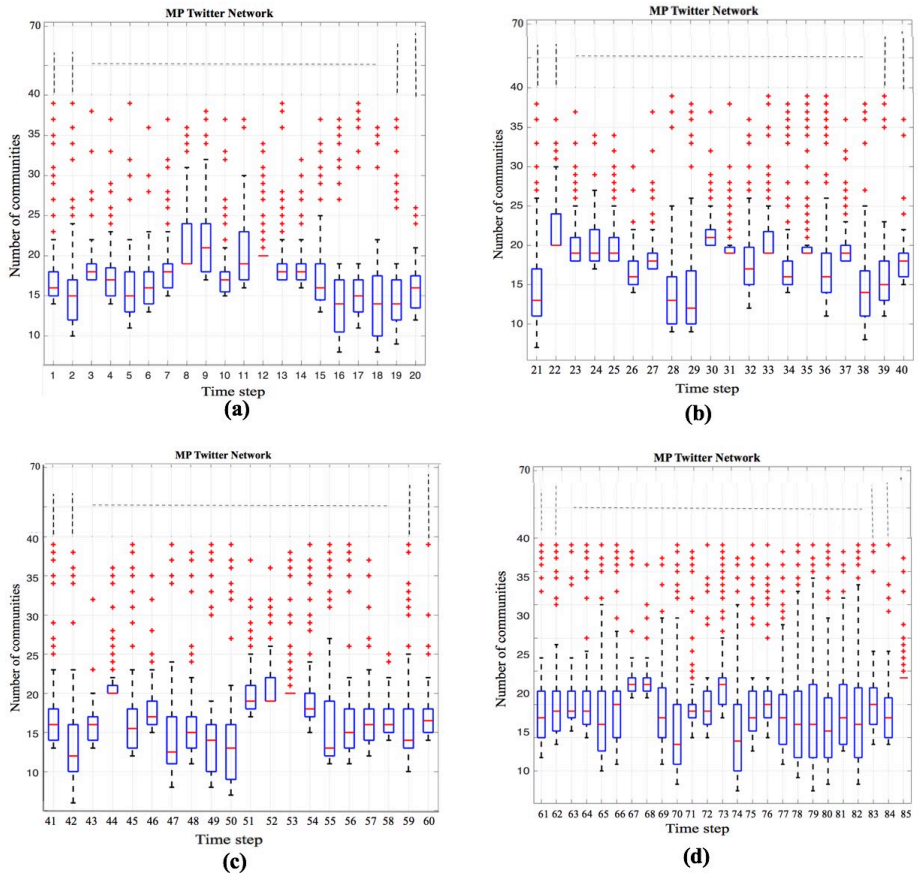


Figure 4.10 The distribution of the number of communities in each of the Pareto optimal solutions found by the MOEA-CD algorithm on 85 weeks of MP Twitter network.

Figure 4.9 shows the development of the communities surrounding the Brexit referendum on 23rd June 2016. Several weeks before the referendum, the network comprises three main communities: Labour, the Conservatives and UKIP, and the Scottish National Party. As the referendum approaches, Conservative, Labour and UKIP MPs merge to form a single large community including both leavers and remainers, but the mainly SNP community remains distinct. Immediately following the referendum, the large community splits again along party lines. Figure 4.10 shows the number of communities for each partition within a set of possible partitions that are found by MOEA algorithm over 85 weeks (time steps). There is considerable variation in the number of communities in partitions found by the MOEA at each timestep. However, the average number of communities is generally between 15 and 20. Despite the fact that most partitions have many possible communities, the usual structure is between 3 and 6 large communities corresponding to the major political groupings together with a number of much smaller communities

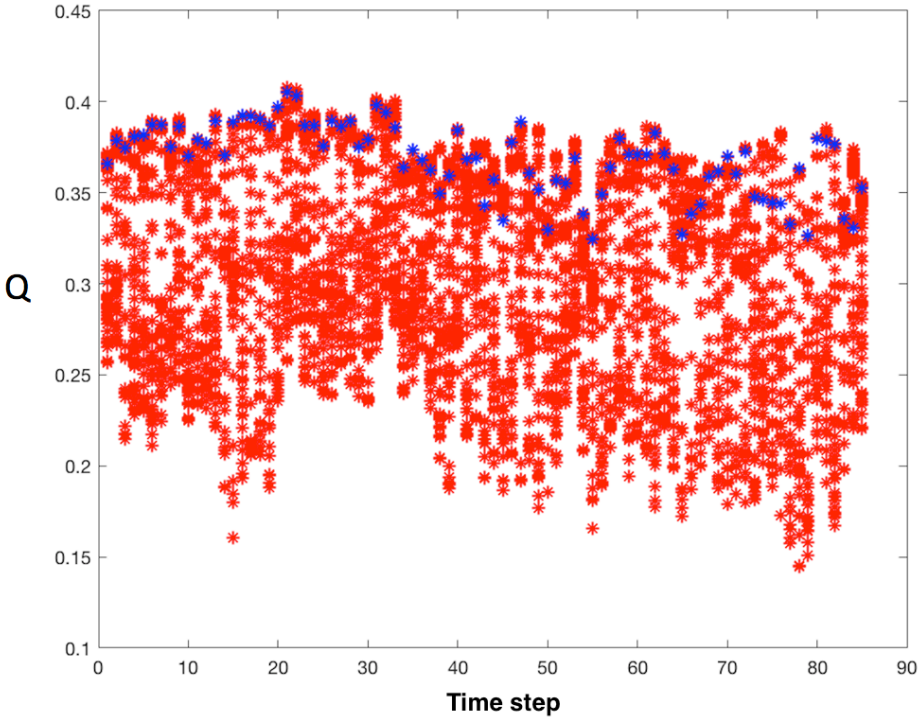


Figure 4.11 Modularity and Viterbi results on MPs Twitter networks. Red stars represent the modularity values for each state (network partition). Blue stars are the most likely sequence of states (network partitions) over 85-time step using the Viterbi algorithm.

corresponding to small groups of a few MPs.

Figure 4.11 shows the modularity value for each state (network partition) in each time step of MPs Twitter networks. The modularity values are represented by red stars. The blue stars are the most likely sequence of states (network partitions) over 85-time step using the Viterbi algorithm. As we can see that the sequence of partitions located by the Viterbi algorithm does not always correspond to the partition with the largest modularity. This is because the Viterbi algorithm balances the modularity at each time with the probability of a transition to that partition from the partition at the previous time.

## 4.3 Summary

In this chapter, we have presented a new methodology to detect and capture the evolution of community structures over time in networks using a Hidden Markov

Model (HMM). In each time step, community structures are detected using a Multi-Objective Evolutionary Algorithm. This algorithm optimises two contradictory objectives which are derived from the investigation of the node relationship within a community and with the rest of communities. After that, the Viterbi algorithm has located the sequence of partitions that are true or closest to the true evolving community structure. However our algorithm doesnt include the similarity measure through optimisation using MOEA to generate Pareto optimal solutions. We rely on Viterbi algorithm to choose the more similar partition over time. In this case, the generated partitions at this stage could lack of the more similar partitions. The results show that our algorithm is effective and promising. It seems more likely to understand the relations of each node with other nodes within the community and with the rest, provides more opportunity to understand the evolution of communities over time. The proposed algorithm is presented for unweighted and undirected networks. However, in the real world, there are many weighted and directed networks that could be suggested as future research. In addition, we can develop our algorithm to detect complex models in Protein-Protein Interaction Networks. These networks are different from social networks because the size of communities in biological networks is generally smaller than the size of communities in social networks, while the number of communities in biological networks is larger than the number of communities in social networks.

# Chapter 5

# Conclusion and Future Work

In this chapter, we describe the main contributions of this thesis and then point to the further future directions that could be extended by this thesis.

## 5.1 Summary of Contributions

In the last one and half decades, community analysis of complex networks has become an important research topic. Much of the work is devoted for community analysis in the networks. Some of them have a good contribution to discover communities. However, they still suffer from accuracy limitations in term of identifying the structure of communities when they evaluated different real-life networks. These networks have different structural properties. As a result, the question of discovering community structures is still open. Therefore we started our investigation by studying the smallest level node relationships with its neighbour nodes. Then we studied the scores that have been proposed to capture the best network partition into clusters. Following this, we proposed an algorithm for community detection in static networks. Finally, this algorithm is developed to capture the evolution of communities in dynamic networks.

In the following sections, we draw our main contributions in this thesis separately for community analysing in the network systems.

### 5.1.1 Evaluation of Community Scores

The score functions represent the heart of any optimisation algorithm for community detection as these algorithms use the score functions to evaluate the network partition. These scores quantify how well a particular network partition fits a given network. Therefore it is an exciting study to find a strategy to evaluate these scores.

Without loss of generality, suppose that an objective is to be maximised, then the goodness score is achieved if its value is the largest value when it is evaluated on the correct partition. We assess community scores on five real networks. The results showed that some of the score functions have a good performance to evaluate network partitions while others need more improvement. This study is very important to evaluate the score quality before optimisation and show the correlation between the objective and $NMI$. However, the difficulty in this study that the ground-truth partition may not be reliable, particularly when the ground-truth partition has weak nodes. In addition, the randomly generated partitions could lack the structures of a natural network partition (dense connections within the community and few connections with others communities) as there are no constraints on which nodes will be selected. For example, the selected strong nodes could become weak nodes in other communities.

On the other hand, the ground-truth partitions are more similar to the natural partitions in term of community structures (dense connections within the community and sparse connections with others) as these partitions are created in real-world networks. Small perturbations of the true partitions represent a useful set of partitions for objective evaluation as these partitions are more structures than the large perturbations

### 5.1.2 Community Detection in Static Networks

We introduce two new objectives Intra-score and Inter-score to evaluate the goodness of network partitions by optimising these two objectives using a Multi-Objective Evolutionary Algorithm presenting a new Multi-Objective Evolutionary Algorithm for Community Detection (MOEA-CD). The objectives are derived from our investigation into the node relationship within a community and with the rest of communities. The first objective (Intra-score) is to increase the number of connections inside communities while the second one (Inter-score) is to decrease the number of connections between communities.

We can see from our results for evaluating the performance of the scores in chapter three that these two objectives have a good correlation with $NMI$. Moreover, a new local heuristic search method based on Neighbour Node Centrality definition is combined with our algorithm to speed up the converge of MOEA-CD to an optimal solution. As mentioned before, this heuristic procedure has a positive impact on optimising the objectives of four models on both synthetic and real-world networks. However, there is additional time complexity due to this process. From our experiments, we have observed that our algorithm is effective and promising by investigating its performance in comparison to three state-of-the-art models with and without the local heuristic search on 28 real-world and synthetic networks. We believe that MOEA-CD can produce more accurate community structure than others because it concentrates on node relationships. However, one of the limitations in this algorithm is the time complexity compared with the traditional optimisation algorithms. The execution time for an evolutionary algorithm is dominated by the calculation of the objective function specifically when the population size is large when using the multi-objective algorithm.

There is another difficulty with using an MOEA for community detection. We know that MOEAs generate a set of candidate solutions or partitions, so the difficulty in choosing the best partition among these available candidate partitions, i.e. the best

partition means the community structure where each community contains nodes that share specific activity. We used modularity in our study to select the best partition and it is a good measure but it is not perfect.

### 5.1.3 Community Detection in Dynamic Networks

We have developed our algorithm (MOEA-CD) to analyse the evolution of communities over time using a Hidden Markov Model. MOEA-CD is used to generate many possible states which represent Pareto-optimal solutions (network partitions) at each time step. Then the Viterbi algorithm is used to find the most likely sequence of partitions over time. The performance of this method has been assessed on the synthetic and real-world network. The results showed that our algorithm is successful in simultaneously detecting accurate community structures at each time step and similarity between successive time steps. However the problem in the number of possible states at each time step. As we are mentioned it is impossible to consider all possible network partitions as they are huge. Therefore, our algorithm is still constrained by the quality of given partitions (hidden states). The possible solutions that could be done to minimise this limitation, for example, keeping some of the solutions behind the Pareto front or starting the MOEA from the solutions found at the previous timestep, etc. Also, note that this is likely to be a problem for any dynamic algorithm.

## 5.2 Future work

In this section, we offer suggestions for future research directions and open question that extend our study.

### 5.2.1 Community Score Evaluation

Further future work can be followed by our strategy for evaluating the score functions. As the number of partitions that are used for evaluating the quality of the score function is large, the evaluation method will be more accurate. However, the number of possible partitions for a given network is huge. Therefore, it is very interesting to find another method that could generate another set of network partitions. These partitions could be combined with our set of partitions which are generated by random migration strategy. We plan to improve our random migration strategy to include more perturbations of the network partition and investigate how the new set of network partitions is the effect on the behaviour of community scores. One of the possible methods that could generate useful network partitions is Single Objective Evolutionary Algorithm and other different optimisation algorithms.

In addition, Metropolis-Hastings algorithm [Hastings, 1970] could be used to generate another sequence of a random sample of network partitions that could be combined with the above sets to produce a variety of network partitions where the score functions are evaluated on them. As we discussed earlier, the ground-truth partition is unreliable. Therefore, it will be an interesting study if a gold standard partition is generated based on the ground-truth partition. At least the weak nodes that exist in the true partition should be removed from the ground-truth partition to generate gold standard partition. In addition, constraints are needed on the selected nodes to keep the intuition of community structures (more connections within the community and few with other communities).

### 5.2.2 Community Detection in Static Networks

It seems more likely to understand the relations of each node with other nodes within the community and with the rest communities as that provide more opportunity to understand the structure of communities. The networks investigated in this thesis

are unweighted, undirected and unsigned. It is more interesting if we extend our study on these different types of networks. The communities in signed networks are detected by density and signs as well as the links between nodes. That means the links are positive and negative between nodes in these networks. In this case, the nodes that have negative relations with the neighbour nodes, they may be assigned to different communities. Therefore, these relations between nodes need more investigation to detect the accurate and fast community structures at the same time in signed networks. From this point, we plan to develop and harness our objective functions and heuristic strategy for detecting the community structures in these networks.

We also are interested in investigating how to select the best solution in Pareto front when the true partition is unknown.

### 5.2.3 Community Detection in Dynamic Networks

As we discussed earlier that we could not directly apply filtering and smoothing algorithms based on HMM as the number of possible states is vast. Therefore the likely venue for future direction in detecting communities in dynamic networks is how to find another possible set of possible states (network partitions) at each time step. Dirichlet processes can be used to represent HMM with a very large number of hidden state. In addition, it could generate useful network partitions is Single Objective Evolutionary Algorithm and other different optimisation algorithms to combined with the set of possible states at each time step in the dynamic algorithm.

Another direction to investigate our algorithm to analyse the evolution of protein-protein interaction networks. In these networks, the proteins represent nodes and edges represent the interaction between the two proteins. These interactions are changed with the change of protein's age, and that will change the biological functions. It is essential to understand the evolution of these interactions over time.

# Bibliography

Abido, M. (2003). A Niched Pareto Genetic Algorithm for Multiobjective Environmental/Economic Dispatch. *International Journal of Electrical Power & Energy Systems*, 25(2):97–105.

Agarwal, G. and Kempe, D. (2008). Modularity-Maximizing Graph Communities via Mathematical Programming. *The European Physical Journal B*, 66(3):409–418.

Albert, R. k. and Barabási, A.-L. (2002). Statistical Mechanics of Complex Networks. *Reviews of Modern Physics*, 74(1):47.

Amigó, E., Gonzalo, J., Artiles, J., and Verdejo, F. (2009). A Comparison of Extrinsic Clustering Evaluation Metrics based on Formal Constraints. *Information Retrieval*, 12(4):461–486.

Amini, A. A., Chen, A., Bickel, P. J., Levina, E., et al. (2013). Pseudo-Likelihood Methods for Community Detection in Large Sparse Networks. *The Annals of Statistics*, 41(4):2097–2122.

Angelini, L., Boccaletti, S., Marinazzo, D., Pellicoro, M., and Stramaglia, S. (2007). Identification of Network Modules by Optimization of Ratio Association. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 17(2):023114.

Bäck, T., Fogel, D. B., and Michalewicz, Z. (1997). *Handbook of Evolutionary Computation*. CRC Press.

Bader, J. and Zitzler, E. (2011). HypE: An Algorithm for Fast Hypervolume-based Many-Objective Optimization. *Evolutionary Computation*, 19(1):45–76.

Banavar, J. R., Colaiori, F., Flammini, A., Maritan, A., and Rinaldo, A. (2000). Topology of the Fittest Transportation Network. *Physical Review Letters*, 84(20):4745.

Bickel, P. J. and Chen, A. (2009). A Nonparametric View of Network Models and Newman–Girvan and Other Modularities. *Proceedings of the National Academy of Sciences*, pages pnas–0907096106.

Blondel, V., Guillaume, J.-L., Lambiotte, R., and Lefebvre, E. (2008). Fast Unfolding of Communities in Large Networks. *J. Stat. Mech., P10008*.

Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press.

Brandes, U. (2001). A Faster Algorithm for Betweenness Centrality. *Journal of Mathematical Sociology*, 25(2):163–177.

Brandes, U., Delling, D., Gaertler, M., Gorke, R., Hoefer, M., Nikoloski, Z., and Wagner, D. (2008). On Modularity Clustering. *IEEE Transactions on Knowledge and Data Engineering*, 20(2):172–188.

Broder, A., Kumar, R., Maghoul, F., Raghavan, P., Rajagopalan, S., Stata, R., Tomkins, A., and Wiener, J. (2000). Graph Structure in The Web. *Computer Networks*, 33(1):309–320.

Cazabet, R. and Amblard, F. (2014). Dynamic Community Detection. In *Encyclopedia of Social Network Analysis and Mining*, pages 404–414. Springer.

Chakrabarti, D., Kumar, R., and Tomkins, A. (2006). Evolutionary Clustering. In *Proceedings of The 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 554–560. ACM.

Chang, P. C., Chen, S. H., Zhang, Q., and Lin, J. L. (2008). MOEA/D for Flowshop Scheduling Problems. In *Evolutionary Computation, 2008. CEC 2008.(IEEE World Congress on Computational Intelligence). IEEE Congress on*, pages 1433–1438. IEEE.

Chen, M., Kuzmin, K., and Szymanski, B. K. (2014). Community Detection Via Maximization of Modularity and Its Variants. *IEEE Transactions on Computational Social Systems*, 1(1):46–65.

Cheng, F., Cui, T., Su, Y., Niu, Y., and Zhang, X. (2018). A Local Information Based Multi-Objective Evolutionary Algorithm for Community Detection in Complex Networks. *Applied Soft Computing*, 69:357–367.

Cherkassky, B. V., Goldberg, A. V., and Radzik, T. (1996). Shortest Paths Algorithms: Theory and Experimental Evaluation. *Mathematical programming*, 73(2):129–174.

Clauset, A., Newman, M. E., and Moore, C. (2004). Finding Community Structure In Very Large Networks. *Physical Review E*, 70(6):066111.

Coello, C. A. C., Lamont, G. B., and Van Veldhuizen, D. A. (2007). *Evolutionary Algorithms for Solving Multi-Objective Problems*, volume 5. New York: Springer.

Corne, D. W. and Lones, M. A. (2018). Evolutionary Algorithms. *arXiv preprint arXiv:1805.11014*.

Danon, L., Díaz-Guilera, A., and Arenas, A. (2006). The Effect of Size Heterogeneity on Community Identification in Complex Networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2006(11):P11010.

Danon, L., Diaz-Guilera, A., Duch, J., and Arenas, A. (2005). Comparing Community Structure Identification. *Journal of Statistical Mechanics: Theory and Experiment*, 2005(09):P09008.

Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II . *Evolutionary Computation, IEEE Transactions on*, 6(2):182–197.

Dhillon, I. S., Guan, Y., and Kulis, B. (2004). *A Unified View of Kernel K-Means, Spectral Clustering and Graph Cuts*. University of Texas.

Donath, W. E. and Hoffman (1973). Lower Bounds for the Partitioning of Graphs. *IBM Journal of Research and Development*, 17(5):420–425.

Duan, L., Street, W. N., Liu, Y., and Lu, H. (2014). Community Detection in Graphs Through Correlation. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1376–1385. ACM.

Emmerich, M., Beume, N., and Naujoks, B. (2005). An EMO Algorithm using the Hypervolume Measure as Selection Criterion. In *International Conference on Evolutionary Multi-Criterion Optimization*, pages 62–76. Springer.

Faust, K. (1997). Centrality in Affiliation Networks. *Social Networks*, 19(2):157–191.

Folino, F. and Pizzuti, C. (2010). A Multiobjective and Evolutionary Clustering Method for Dynamic Networks. In *Advances in Social Networks Analysis and Mining (ASONAM), 2010 International Conference on*, pages 256–263. IEEE.

Folino, F. and Pizzuti, C. (2014). An Mvolutionary Multiobjective Approach for Community Discovery in Dynamic Networks. *IEEE Transactions on Knowledge and Data Engineering*, 26(8):1838–1852.

Fonseca, C. M. and Fleming, P. J. (1996). On the Performance Assessment and Comparison of Stochastic Multiobjective Optimizers. In *International Conference on Parallel Problem Solving from Nature*, pages 584–593. Springer.

Fonseca, C. M., Fleming, P. J., et al. (1993). Genetic Algorithms for Multiobjective

Optimization: Formulation Discussion and Generalization. In *ICGA*, volume 93, pages 416–423.

Fortunato, S. and Barthelemy, M. (2007). Resolution Limit in Community Detection. *Proceedings of the National Academy of Sciences*, 104(1):36–41.

Fortunato, S. and Hric, D. (2016). Community Detection in Networks: A User Guide. *Physics Reports*, 659:1–44.

Fortunato, S. and Lancichinetti, A. (2009). Community Detection Algorithms: A Comparative Analysis: Invited Presentation, Extended Abstract. In *Proceedings of the Fourth International ICST Conference on Performance Evaluation Methodologies and Tools*, page 27. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

Freeman, L. C. (1977). A Set of Measures of Centrality based on Betweenness. *Sociometry*, pages 35–41.

Fruchterman, T. M. and Reingold, E. M. (1991). Graph Drawing by Force-Directed Placement. *Software: Practice and Experience*, 21(11):1129–1164.

Girvan, M. and Newman, M. E. (2002). Community structure in Social and Biological Networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826.

Gleiser, P. M. and Danon, L. (2003). Community Structure in Jazz. *Advances in Complex sSystems*, 6(04):565–573.

Goldberg, D. E. and Holland, J. H. (1988). Genetic Algorithms and Machine Learning. *Machine learning*, 3(2):95–99.

Gong, M., Cai, Q., Chen, X., and Ma, L. (2014). Complex Network Clustering by Multiobjective Discrete Particle Swarm Optimization Based on Decomposition. *Evolutionary Computation, IEEE Transactions on*, 18(1):82–97.

Greene, D., Doyle, D., and Cunningham, P. (2010). Tracking the Evolution of Communities in Dynamic Social Networks. In *Advances in Social Networks Analysis and Mining (ASONAM), 2010 International Conference on*, pages 176–183. IEEE.

Grinstein, G., Plaisant, C., Laskowski, S., O'connell, T., Scholtz, J., and Whiting, M. (2008). VAST 2008 Challenge: Introducing Mini-Challenges. In *Visual Analytics Science and Technology, 2008. VAST'08. IEEE Symposium on*, pages 195–196. IEEE.

Guimera, R. and Amaral, L. A. N. (2005). Cartography of Complex Networks: Modules and Universal Roles. *Journal of Statistical Mechanics: Theory and Experiment*, 2005(02):P02001.

Guimera, R., Sales-Pardo, M., and Amaral, L. A. N. (2004). Modularity from Fluctuations in Random Graphs and Complex Networks. *Physical Review E*, 70(2):025101.

Hafez, A. I., Al-Shammari, E. T., ella Hassanien, A., and Fahmy, A. A. (2014). Genetic algorithms for multi-objective community detection in complex networks. In *Social Networks: A Framework of Computational Intelligence*, pages 145–171. Springer.

Handl, J. and Knowles, J. (2007). An Evolutionary Approach to Multiobjective Clustering. *Evolutionary Computation, IEEE Transactions on*, 11(1):56–76.

Hariz, W. A., Abdulhalim, M. F., et al. (2016). Improving the performance of evolutionary multi-objective co-clustering models for community detection in complex social networks. *Swarm and Evolutionary Computation*, 26:137–156.

Hastings, W. K. (1970). Monte Carlo Sampling Methods using Markov Chains and their Applications. *Biometrika*, 57:97–109.

Hofman, J. M. and Wiggins, C. H. (2008). Bayesian Approach to Network Modu-

larity. *Physical Review Letters*, 100(25):258701.

Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems: An Introductory Analysis With Applications to Biology, Control, and Artificial Intelligence.* University of Michigan Press Ann Arbor.

Holland, P. W., Laskey, K. B., and Leinhardt, S. (1983). Stochastic Blockmodels: First Steps. *Social Networks*, 5(2):109–137.

Hopcroft, J., Khan, O., Kulis, B., and Selman, B. (2004). Tracking Evolving Communities in Large Linked Networks. *Proceedings of The National Academy of Sciences*, 101(suppl 1):5249–5253.

Hopfe, C. J. (2009). Uncertainty and Sensitivity Analysis in Building Performance Simulation for Decision Support and Design Optimization. *PhD diss., Eindhoven University.*

Hughes, B. D. (1995). *Random Walks and Random Environments*, volume 1. Clarendon Press; Oxford University Press.

Hughes, E. J. (2007). MSOPS-II: A general-Purpose Many-Objective Optimiser. In *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on*, pages 3944–3951. IEEE.

Ishibuchi, H., Sakane, Y., Tsukamoto, N., and Nojima, Y. (2009). Evolutionary Many-Objective Optimization by NSGA-II and MOEA/D with Large Populations. In *Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on*, pages 1758–1763. IEEE.

Ishibuchi, H., Sakane, Y., Tsukamoto, N., and Nojima, Y. (2010). Simultaneous Use of Different Scalarizing Functions in MOEA/D. In *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation*, pages 519–526. ACM.

Ishibuchi, H., Tsukamoto, N., and Nojima, Y. (2008). Evolutionary Many-Objective

Optimization. In *Genetic and Evolving Systems, 2008. GEFS 2008. 3rd International Workshop on*, pages 47–52. IEEE.

Jaccard, P. (1908). Nouvelles Recherches Sur la Distribution Florale. *Bull. Soc. Vaud. Sci. Nat.*, 44:223–270.

Jain, A. K. and Dubes, R. C. (1988). *Algorithms for Clustering Data*. Upper Saddle River, NJ: Prentice-Hall, Inc.

Jain, A. K., Murty, M. N., and Flynn, P. J. (1999). Data Clustering: A Review. *ACM Computing Surveys (CSUR)*, 31(3):264–323.

Jaszkiewicz, A. (2004). On the Computational Efficiency of Multiple Objective Metaheuristics. The knapsack Problem Case Study. *European Journal of Operational Research*, 158(2):418–433.

Karrer, B. and Newman, M. E. (2011). Stochastic Blockmodels and Community Structure in Networks. *Physical Review E*, 83(1):016107.

Kim, M.-S. and Han, J. (2009). A Particle-and-Density Based Evolutionary Clustering Method for Dynamic Networks. *Proceedings of the VLDB Endowment*, 2(1):622–633.

Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P. (1983). Optimization by Simulated Annealing. *Science*, 220(4598):671–680.

Knowles, J. D. and Corne, D. W. (2000). Approximating the Nondominated Front using the Pareto Archived Evolution Strategy. *Evolutionary Computation*, 8(2):149–172.

Konak, A., Coit, D. W., and Smith, A. E. (2006). Multi-Objective Optimization Using Genetic Algorithms: A Tutorial. *Reliability Engineering & System Safety*, 91(9):992–1007.

Konstantinidis, A. and Yang, K. (2011). Multi-Objective Energy-Efficient Dense Deployment in Wireless Sensor Networks Using A Hybrid Problem-Specific MOEA/D. *Applied Soft Computing*, 11(6):4117–4134.

Konstantinidis, A., Yang, K., Zhang, Q., and Zeinalipour-Yazti, D. (2010). A Multi-Objective Evolutionary Algorithm for the Deployment and Power Assignment Problem in Wireless Sensor Networks. *Computer Networks*, 54(6):960–976.

Kuhn, H. W. (1955). The Hungarian Method for The Assignment Problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97.

Kumar, R., Novak, J., Raghavan, P., and Tomkins, A. (2005). On The Bursty Evolution of Blogspace. *World Wide Web*, 8(2):159–178.

Lancichinetti, A. and Fortunato, S. (2009). Community Detection Algorithms: a Comparative Analysis. *Physical Review E*, 80(5):056117.

Lancichinetti, A. and Fortunato, S. (2011). Limits of Modularity Maximization in Community Detection. *Physical Review E*, 84(6):066122.

Lancichinetti, A. and Fortunato, S. (2012). Consensus Clustering in Complex Networks. *Scientific Reports*, 2:336.

Lancichinetti, A., Fortunato, S., and Kertész, J. (2009). Detecting the Overlapping and Hierarchical Community Structure in Complex Networks. *New Journal of Physics*, 11(3):033015.

Lancichinetti, A., Fortunato, S., and Radicchi, F. (2008). Benchmark Graphs for Testing Community Detection Algorithms. *Physical review E*, 78(4):046110.

Lancichinetti, A., Kivelä, M., Saramäki, J., and Fortunato, S. (2010). Characterizing the Community Structure of Complex Networks. *PloS One*, 5(8):e11976.

Lancichinetti, A., Radicchi, F., Ramasco, J. J., and Fortunato, S. (2011). Finding Statistically Significant Communities in Networks. *PloS one*, 6(4):e18961.

Lee, J. and Lee, J. (2013). Hidden Information Revealed by Optimal Community Structure from A Protein-Complex Bipartite Network Improves Protein Function Prediction. *PloS One*, 8(4):e60372.

Leskovec, J., Kleinberg, J., and Faloutsos, C. (2005). Graphs Over Time: Densification Laws, Shrinking Diameters and Possible Explanations. In *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, pages 177–187. ACM.

Li, H. and Zhang, Q. (2006). A Multiobjective Differential Evolution based on Decomposition for Multiobjective Optimization with Variable Linkages. In *Parallel Problem Solving from Nature-PPSN IX*, pages 583–592. Springer.

Li, H. and Zhang, Q. (2009). Multiobjective Optimization Problems with Complicated Pareto Sets, MOEA/D and NSGA-II. *IEEE Transactions on Evolutionary Computation*, 13(2):284–302.

Li, Z., Zhang, S., Wang, R.-S., Zhang, X.-S., and Chen, L. (2008). Quantitative Function for Community Detection. *Physical Review E*, 77(3):036109.

Lin, Y.-R., Chi, Y., Zhu, S., Sundaram, H., and Tseng, B. L. (2009). Analyzing Communities and Their Evolutions in Dynamic Social Networks. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 3(2):8.

Lusseau, D. (2003). The Emergent Properties of A Dolphin Social Network. *Proceedings of The Royal Society of London B: Biological Sciences*, 270(Suppl 2):S186–S188.

Ma, J., Liu, J., Ma, W., Gong, M., and Jiao, L. (2014). Decomposition-Based Mul-

tiobjective Evolutionary Algorithm for Community Detection in Dynamic Social Networks. *The Scientific World Journal*, 2014(402345).

Massen, C. P. and Doye, J. P. (2005). Identifying Communities within Energy Landscapes. *Physical Review E*, 71(4):046101.

Miyauchi, A. and Kawase, Y. (2016). Z-Score-Based Modularity for Community Detection in Networks. *PloS One*, 11(1):e0147805.

Nadler, B. and Galun, M. (2007). Fundamental Limitations of Spectral Clustering. In *Advances in Neural Information Processing Systems*, pages 1017–1024.

Newman, M. (2018). *Networks*. Oxford University Press.

Newman, M. E. (2001). The Structure of Scientific Collaboration Networks. *Proceedings of the National Academy of Sciences*, 98(2):404–409.

Newman, M. E. (2006). Modularity and Community Structure in Networks. *Proceedings of the National Academy of Sciences*, 103(23):8577–8582.

Newman, M. E. (2008). The Mathematics of Networks. *The New Palgrave Encyclopedia of Economics*, 2(2008):1–12.

Newman, M. E. and Girvan, M. (2004). Finding and Evaluating Community Structure in Networks. *Physical review E*, 69(2):026113.

Newman, M. E. and Leicht, E. A. (2007). Mixture Models and Exploratory Analysis in Networks. *Proceedings of the National Academy of Sciences*, 104(23):9564–9569.

Ng, A. Y., Jordan, M. I., and Weiss, Y. (2002). On Spectral Clustering: Analysis and an Algorithm. In *Advances in Neural Information Processing Systems*, pages 849–856.

Nguyen, N. P., Dinh, T. N., Shen, Y., and Thai, M. T. (2014). Dynamic Social Community Detection and Its Applications. *Plos One*, 9(4):e91431.

Palla, G., Barabási, A.-L., and Vicsek, T. (2007). Quantifying Social Group Evolution. *Nature*, 446(7136):664.

Park, Y. and Song, M. (1998). A Genetic Algorithm for Clustering Problems. In *Proceedings of the Third Annual Conference on Genetic Programming*, volume 1998, pages 568–575.

Peng, W., Zhang, Q., and Li, H. (2009). Comparison Between MOEA/D and NSGA-II on the Multi-Objective Travelling Salesman Problem. In *Multi-Objective Memetic Algorithms*, pages 309–324. Springer.

Pizzuti, C. (2008). GA-Net: A Genetic Algorithm for Community Detection in Social Networks. In *International Conference on Parallel Problem Solving from Nature*, pages 1081–1090. Springer.

Pizzuti, C. (2012). A Multiobjective Genetic Algorithm to Find Communities in Complex Networks. *Evolutionary Computation, IEEE Transactions on*, 16(3):418–430.

Pons, P. and Latapy, M. (2006). Computing Communities in Large Networks using Random walks. *J. Graph Algorithms Appl.*, 10(2):191–218.

Preparata, F. P., Wu, X., and Yin, J. (2008). *Frontiers in Algorithmics: Second International Workshop, FAW 2008, Changsha, China, June 19-21, 2008, Proceedings*, volume 5059. Springer Science & Business Media.

Rabiner, L. R. (1989). A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings of the IEEE*, 77(2):257–286.

Radicchi, F., Castellano, C., Cecconi, F., Loreto, V., and Parisi, D. (2004). Defining

and Identifying Communities in Networks. *Proceedings of the National Academy of Sciences*, 101(9):2658–2663.

Schaffer, J. D. (1985). Multiple Objective Optimization with Vector Evaluated Genetic Algorithms. In *Proceedings of the First International Conference on Genetic Algorithms and Their Applications, 1985*. Lawrence Erlbaum Associates. Inc., Publishers.

Schütze, H., Manning, C. D., and Raghavan, P. (2008). *Introduction to Information Retrieval*, volume 39. Cambridge University Press.

Scott, J. (2017). *Social Network Analysis: A Handbook*. Sage Publications, London, 4nd edition.

Shi, C., Yan, Z., Cai, Y., and Wu, B. (2012). Multi-Objective Community Detection in Complex Networks. *Applied Soft Computing*, 12(2):850–859.

Shi, C., Yu, P. S., Yan, Z., Huang, Y., and Wang, B. (2014). Comparison and Selection of Objective Functions in Multiobjective Community Detection. *Computational Intelligence*, 30(3):562–582.

Shi, J. and Malik, J. (2000). Normalized Cuts and Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905.

Srinivas, N. and Deb, K. (1994). Muiltiobjective Optimization using Nondominated Sorting in Genetic Algorithms. *Evolutionary Computation*, 2(3):221–248.

Sun, P. G. and Sun, X. (2017). Complete Graph Model for Community Detection. *Physica A: Statistical Mechanics and Its Applications*, 471:88–97.

Tang, L., Liu, H., Zhang, J., and Nazeri, Z. (2008). Community Evolution in Dynamic Multi-Mode Networks. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 677–685. ACM.

Van Rijsbergen, C. J. (1979). *Information Retrieval (2nd ed.)*. London: Butterworth.

Varadarajan, M. and Swarup, K. S. (2008). Solving Multi-Objective Optimal Power Flow using Differential Evolution. *IET Generation, Transmission & Distribution*, 2(5):720–730.

von Lücken, C., Barán, B., and Brizuela, C. (2014). A Survey on Multi-Objective Evolutionary Algorithms for Many-Objective Problems. *Computational Optimization and Applications*, 58(3):707–756.

Von Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416.

Wang, X. F. and Chen, G. (2003). Complex Networks: Small-World, Scale-Free And Beyond. *IEEE Circuits and Systems Magazine*, 3(1):6–20.

Wasserman, S. and Pattison, P. (1996). Logit Models and Logistic Regressions for Social Networks: I. An Introduction to Markov Graphs andp. *Psychometrika*, 61(3):401–425.

Weaver, I. S., Williams, H., Cioroianu, I., Williams, M., Coan, T., and Banducci, S. (2018). Dynamic social media affiliations among uk politicians. *Social Networks*, 54:132–144.

White, S. and Smyth, P. (2005). A Spectral Clustering Approach to Finding Communities in Graphs. In *Proceedings of the 2005 SIAM International Conference on Data Mining*, pages 274–285. SIAM.

Wu, P. and Pan, L. (2015). Multi-Objective Community Detection Based on Memetic Algorithm. *Plos One*, 10(5):e0126845.

Xu, K. S., Kliger, M., and Hero Iii, A. O. (2014). Adaptive Evolutionary Clustering. *Data Mining and Knowledge Discovery*, 28(2):304–336.

Yang, J. and Leskovec, J. (2015). Defining and Evaluating Network Communities Based on Ground-Truth. *Knowledge and Information Systems*, 42(1):181–213.

Zachary, W. W. (1977). An Information Flow Model for Conflict and Fission in Small Groups. *Journal of Anthropological Research*, pages 452–473.

Zhang, Q. and Li, H. (2007). MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *Evolutionary Computation, IEEE Transactions on*, 11(6):712–731.

Zhang, Q., Liu, W., Tsang, E., and Virginas, B. (2010). Expensive Multiobjective Optimization by MOEA/D with Gaussian Process Model. *IEEE Transactions on Evolutionary Computation*, 14(3):456–474.

Zhao, X., Li, Y., and Qu, Z. (2018). A Density-Based Clustering Model for Community Detection in Complex Networks. In *AIP Conference Proceedings*, volume 1955, page 040161. AIP Publishing.

Zhao, Y. and Karypis, G. (2001). *Criterion Functions for Document Clustering: Experiments and Analysis*. Technical Report TR 01−40, Department of Computer Science, University of Minnesota.

Zhou, A., Qu, B.-Y., Li, H., Zhao, S.-Z., Suganthan, P. N., and Zhang, Q. (2011). Multiobjective Evolutionary Algorithms: A Survey of the State of the Art. *Swarm and Evolutionary Computation*, 1(1):32–49.

Zhou, H. (2003). Distance, Dissimilarity Index, and Network Community Structure. *Physical Review E*, 67(6):061901.

Zhou, X., Liu, Y., Li, B., and Sun, G. (2015). Multiobjective Biogeography Based Optimization Algorithm With Decomposition for Community Detection in Dynamic Networks. *Physica A: Statistical Mechanics and its Applications*, 436:430–442.

Zitzler, E. (1999). *Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications*, volume 63. Ph. D. thesis, Swiss Federal Institute of Technology (ETH) Zurich, Switzerland.

Zitzler, E. and Künzli, S. (2004). Indicator-based Selection in Multiobjective Search. In *International Conference on Parallel Problem Solving from Nature*, pages 832–842. Springer.

Zitzler, E. and Thiele, L. (1999). Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271.