JADAVPUR UNIVERSITY

MASTER DEGREE THESIS

Detection of Overlapping Communities using Multi-Objective Genetic Algorithms

A thesis submitted in fulfillment of the requirements
for the degree of Master of Engineering

in

Computer Science and Engineering

by

Amit Kumar
Exam Roll No.: M4CSE19028
Class Roll No.: 001710502026
Registration No.: 140765 of 2017-2018

Under the Guidance of

Prof. Nirmalya Chowdhury
Department of Computer Science and Engineering
Jadavpur University

Department of Computer Science and Engineering
Faculty of Engineering and Technology
Jadavpur University
Kolkata - 700032

May 25, 2019

Declaration of Authorship

I, Amit Kumar, declare that this thesis titled, " Detection of Overlapping Communities using Multi-Objective Genetic Algorithms" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a masters degree in computer science and engineering at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely of my own work.
- I have acknowledged all main sources of help.

Signed:

Date:

To Whom It May Concern

This is to certify that the thesis entitled " Detection of Overlapping Communities using Multi-Objective Genetic Algorithms" is a bona-fide record of work carried out by Amit Kumar, Examination Roll No.: M4CSE19028, University Registration No.: 140765 of 2017-2018 in partial fulfillment of the requirements for the award of the degree of Master of Engineering in Computer Science and Engineering from the Department of Computer Science and Engineering, Jadavpur University for the academic session 2017-2019. It is understood that by this approval the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn therein but approve the thesis only for the purpose for which it has been submitted.

_____

Prof. Nirmalya Chowdhury
Department of Computer Science and Engineering
Jadavpur University
Kolkata - 700032

_____

Prof. Mahantapas Kundu
Head of the Department
Department of Computer Science and Engineering
Jadavpur University
Kolkata - 700032

_____

Prof. Chiranjib Bhattacharjee
Dean,Faculty of Engineering and Technology
Jadavpur University
Kolkata - 700032

Certificate of Approval

(Only in case the thesis is approved)

The thesis at instance is hereby approved as a creditable study of an engineering subject carried out and presented in a manner satisfactory for its acceptance as a perquisite to the degree for which it has been submitted. It is understood that by this approval the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn therein, but approve this thesis only for the purpose for which it is submitted.

_____

(Sign of Examiner)
Date:

_____

(Sign of Examiner)
Date:

JADAVPUR UNIVERSITY

Abstract

Faculty of Engineering and Technology
Department of Computer Science and Engineering

Master of Engineering

Detection of Overlapping Communities using Multi-Objective Genetic Algorithms

by Amit Kumar

The science of network analysis has gained great achievement in the modeling of complex real-world systems. These complex networks exhibit the existence of different types of community structures. In the real world the complex networks such as social networks, biological networks transportation network, etc. usually exhibit in-homogeneity which results in a different modular structure having high interconnection within themselves and less interconnection with the other modules, commonly called communities. In the real world these communities are not necessarily disjoint. Some of these communities may have some common participation called overlapping nodes and these communities are overlapping communities. Analyzing such communities has gained significant attention and become one of the major research topics in complex networks.

In this thesis, we have developed a method to handle the complex networks with ground truth modular structure. Our proposed method is based on a multi-objective evolutionary algorithm. Since the community detection problem can be transformed into an optimization problem, therefore we have used NSGA-II as a supporting structure consisting of two objective functions. The first objective function 5.6 maximizes the internal edge density while the other one (i.e eq. 5.7) minimizes the external edge density of a community. We have developed a neighbor based strategy for initializing the initial population, mutation operator and updation method. Apart from this, we have also computed fuzzy membership values for each node. These combined approaches capable of solving the community detection problem. We have validated our method against four real-world networks with known communities. We have compared our method with seven benchmark methods based on standard metrics like gNMI and modularity. Experimental results show that our method is capable of producing high gNMI value compared to modularity.

Acknowledgements

I would like to extend my heartfelt gratitude to people who helped to bring this thesis work to complete. First, I would express my deep and sincere gratitude and appreciation to my supervisor Prof. Nirmalya Chowdhury without whose continuous support and encouragement this work would not have been possible. His assistance, valuable suggestions and personal guidance throughout the duration of the project has played a pivotal role, without which I would never have been able to reach this far.

I would also wish to thank Prof. Mahantapas Kundu, Head of the Department of Computer Science and Engineering, Jadavpur University and Prof. Chiranjib Bhattacharjee, Dean, Faculty of Engineering and Technology, Jadavpur University for providing me all the facilities and for their support to the activities of this research.
I am thankful to my parents and my elder brother who have always been my constant source of support and inspiration.
Last, but not the least, I would like to thank all my friends, classmates and all respected teachers for their valuable suggestions and helpful discussions.

Regards,
Amit Kumar
Exam Roll No.: M4CSE19028
University Registration No.: 140765 of 2017-2018
Department of Computer Science and Engineering
Jadavpur University

I dedicate this to my parents for their continuous support throughout my journey!

Contents

List of Figures

List of Tables

List of Abbreviations

GAs	**G**enetic **A**lgorithms
MOEAs	**M**ulti **O**bjective **E**volutionary **A**lgorithms
NMF	**N**on-Negative **M**atrix **F**actorization
NSGA	**N**on-Dominated **S**orting **G**enetic **A**lgorithms
EO	**E**xternal **O**ptimization
CPM	**C**lique **P**erculation **M**ethod

Chapter 1

Introduction to Community Detection

1.1 Background

Complex networks are an important part of the real world. Many systems that run in our real world can be modeled to form networks that may be simple or complex. These networks can be represented into a graph consisting of a set of nodes and a set of edges. A connected pair of nodes called edge. Networks are generally used to represent the relationship between individuals or objects. They are employed in various domains such as computer science, physics, and mathematics to represent different types of complex systems such as social networks [86], transportation networks [84], collaborative network [88], biological networks [87]. For example, in a social network, a person can be represented as a node and his relations with friends and colleagues can be represented by edges. In a road network, nodes are the locations and the paths between locations are the edges. In molecular networks, nodes represent sets of molecules, and edges are conserved molecular interactions. In citation networks, nodes can be the varieties of entities, and edges can be the collaboration between them. Such types of networks are continuously increasing due to the development of new technologies. For example, the application of advanced technology to some regions construct Smart cities which generates many complex networks like traffic network [82], power grid [85], social networks [86], etc. that need to handle properly. Such networks contain very important latent information which needs to be revealed. So data mining plays a very crucial role in discovering hidden patterns or information from such networked data. Community detection is one of the important tasks in the analysis of such networks in which different groups or modules present in the network have to be identified. Such a group or module is called a community. As the definition of a community is not well defined so we adopt the most acceptable definition of community by the researchers as a community is a group of nodes having a high association within themselves but less interaction with the rest of the network. The following Figure 1.1 show the pictorial view of communities. Mining patterns like community on complex networks, the community detection task is to split the network into a number of groups of vertices which are commonly known as communities and these communities are different from each other.

1.2 Motivation

Mining patterns like community on complex networks, the data mining task is to split the network into a number of groups of vertices called communities. In real-world networks some of the vertices have the potential to participate in multiple communities. Such communities are said to be overlapping communities. For example, an individual in a social network usually has multiple relations towards groups like family, friends, and colleagues. For researchers, they may be active in several domains. Such type of communities usually

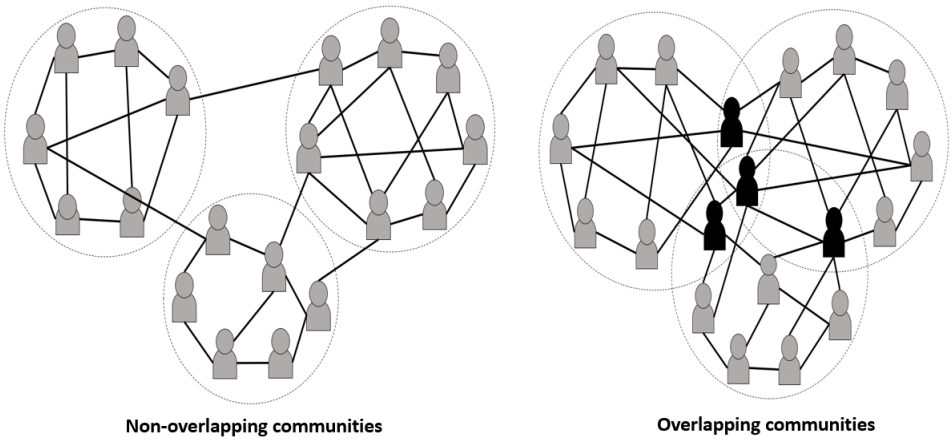

FIGURE 1.1: Non-overlapping vs Overlapping communities.

exists in the real world. Figure 1.1 depicts the overlapping and non-overlapping communities. Community detection for the non-overlapping community is itself a difficult task and if the community becomes overlapping it becomes a great challenge. Over the last decade, overlapping community detection gains significant attention to researchers. Since the community detection problem is NP-Hard so this problem can be converted to an optimization problem and Genetic Algorithm(GA) commonly used to solve such optimization problem. Genetic Algorithm is a stochastic search and the optimization method that uses the concept of natural evaluation. It runs on a population consisting of a set of chromosomes(i.e individual). The individuals of the population are made to pass through the process of evolution (i.e selection, crossover, and mutation) such that fittest individuals go over the next generation or iteration. The applications of such community detection algorithms in the area of social networks, biological network, bibliographic networks, traffic networks, etc. will help us to analyze the topology and functional behavior of network which have great impact on medical science, business, management and many more.

1.3 Aim and Objective

The aim of this thesis is to develop a method that can efficiently identify communities in the context of real-world networks. This requires the development of objective functions that are capable of generating optimal or near optimal community against the ground-truth modular structures which algorithms can be compared against each other. In order to achieve this goal, a number of key points need to be achieved:

- The formal definition community has been provided and the problem of community detection has been modeled as an optimization problem.
- A method suitable for real-world networks with known community structures has been developed.
- The best solution has been chosen based on NMI and Modularity.
- Proposed method has been compared with some existing methods based on NMI and Modularity using real-world networks.

1.4 Contribution

In this section, we have briefly discussed our contribution to the proposed method. We have used NSGA II as working support with objective functions. We have proposed a neighborhood strategy for generating the initial population and fuzzy membership value assignment for each node. We have also proposed a mutation operator having room for both random based and neighbor based strategies while the updation method is used for updating the chromosome after crossover and mutation operators.

1.5 Thesis Organization

The rest of the thesis is organized as follows: Chapter 2 describes the concept of data mining, it's architecture, and different data mining techniques. NSGA-I, and NSGA-II variant of the Genetic Algorithms are described in the Chapter 3. Chapter 4 consists of a literature survey related to community detection algorithms. Our proposed method is discussed in Chapter 5 and the experimental results are presented in Chapter 6. Finally, we conclude and put some light on future work in Chapter 7.

Chapter 2

Data Mining

We are living in a world which is information age world. Information is the basis for running all systems. Due to the great availability of large amounts of data and there is also an immediate need for gathering useful information or knowledge from such data. In past few decades data mining has emerged as a great weapon to handle such huge amount of data. The useful information obtained can be beneficial for us and society. There is wide applications of data mining mention in [72] like market analysis, fraud detection, and customer retention, to production control and science exploration.

Data mining [72] is the process of "extracting" or "mining" useful information from large amounts of data. There are several types of real world mining. For example in coal mining ,we actually mine coal from rocks so the process is termed as coal mining rather than rock mining. The "data mining" term is a misnomer. Information or knowledge mining does not resemble as mining from data but the data mining contain both terms "data" and "mining" that's why it is popular. There are several terms such as knowledge extraction, pattern analysis, data archaeology, data dredging etc. which have similar or very less dissimilar meaning.

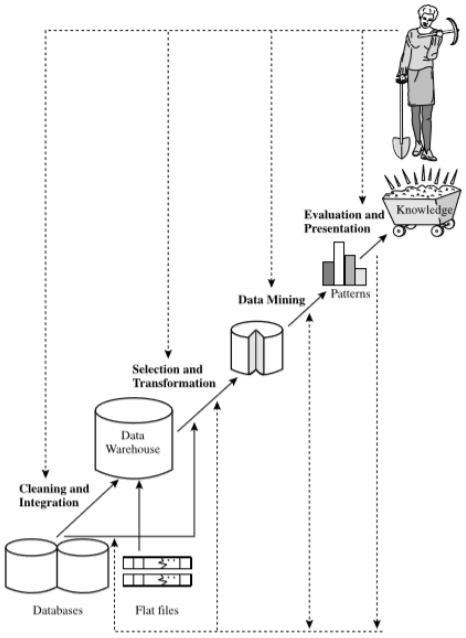

FIGURE 2.1: Data mining steps in the process of knowledge discovery. Source:[72]

1. Data cleaning (to remove noise and inconsistent data)

2. Data integration (where multiple data sources may be combined)

3. Data selection(where data relevant to the analysis task are retrieved from the database)
4. Data transformation(where data are transformed or consolidated into forms appropriate for mining by performing summary or aggregation operations, for instance)
5. Data mining (an essential process where intelligent methods are applied in order to extract data patterns)
6. Pattern evaluation (to identify the truly interesting patterns representing knowledge based on some interestingness measures)
7. Knowledge presentation (where visualization and knowledge representation techniques are used to present the mined knowledge to the user)

2.1 Architecture of a typical data mining system

Based on this view, the architecture of a typical data mining system [72] may have the following major components:

- Database, data warehouse, World Wide Web, or other information repository: This is information storage source in the form of a set of databases, data warehouses, spread sheets, etc. Data cleaning and data integration techniques may be applied on the data.
- Database or data warehouse server: The task of database or data warehouse server is to fetch the relevant data for mining, based on the user's request.
- Knowledge base : This plays a very important role for mining data. This provides domain knowledge to guide the search or evaluate the interestingness of resulting patterns. The domain knowledge can include concept of hierarchical approaches, used to organize attributes or attribute values in to different levels of abstraction. It may also include knowledge based on user beliefs. Apart from this , knowledge base can be used to build various interesting constraints and threshold, and meta data (e.g., describing data from multiple heterogeneous sources.
- Data mining engine: This is essential to the data mining system and ideally consists of a set of functional modules for tasks such as characterization, association and correlation analysis, classification, prediction, cluster analysis, outlier analysis, and evolution analysis.
- Pattern evaluation module: This part uses different interestingness measures to interacts with the data mining modules. Its aim is to the search interesting patterns may using thresholds to discover patterns.
- User interface:This part interacts with the users and the data mining system. It allows users to interact with the system by providing a query related to the data mining providing helpful information to search, and performing exploratory data mining based on the intermediate data mining results.

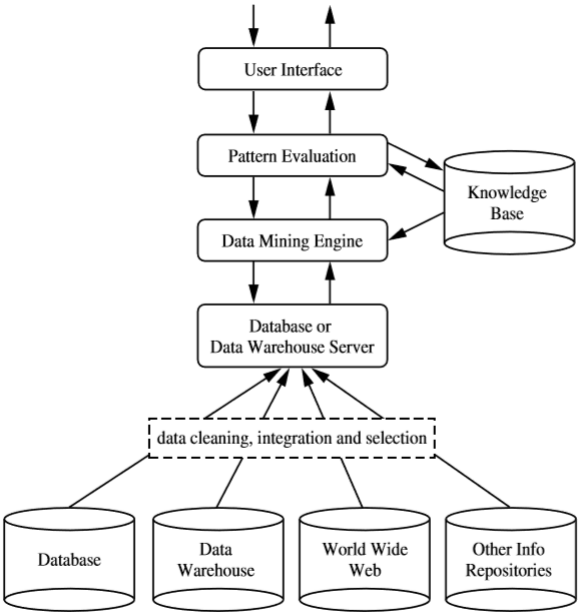

FIGURE 2.2: Architecture of a typical datamining process. Source:[72]

The tasks of data mining are mainly categorized into two forms.

1. Predictive mining : The aim of predictive mining is to predict unknown or future results instead of current behavior. The predictive analysis is used to present information about "what might happen ?" and "why it might happen ?" to the data.

2. Descriptive mining : Descriptive mining is commonly used to find the regularities in data to discover interesting, human understandable patterns. It focuses on the summarization and conversion of data into meaningful information. Descriptive analysis is used to provide information about "what has happend ?" and "what is happening ?" in the data.

2.2 Data Mining Techniques/Tools

Data mining is an interdisciplinary field that includes the different diversity of problems. Therefore the integration of tools/techniques from such areas should be incorporated. Following are the several data mining tools which are widely used to solve data mining problem.

- Soft computing
- Machine learning
- Statistics
- Rough set theory,
- Database Technology
- Different visualization tools

2.2.1 Soft Computing

The concept of Soft computing was introduced by Professor Lotfi Zadeh in nearly mid 90's with an aim to handle the imprecision, uncertainty and partial truth. This concept was developed for better understanding with reality in order to achieve tractability, robustness, low solution cost. The ultimate objective is to closely match with human mind. For example, to park a car parking slot having rectangular line margin. Our goal is to successfully park the vehicle(car) in given parking slot. In reality, we don't bother about how perfectly fit in that slot from each wheel position equally in left and right direction. We generally deal with approximation to save time and cost. Human mind is able to handle such approximation in order to successfully achieve a goal. Soft computing was developed to handle such approximation as human mind does. So it comprises several domain like neural network,fuzzy logic and genetic algorithms. It also includes probabilistic reasoning. The important thing is that the involvement of different domains are complementary not competitive.

Soft Computing Tools

There are generally three main tools/techniques used in soft computing.These are Neural network, Genetic Algorithm and Fuzzy logic.

A neural network is a processing system which can be a hardware device or an algorithm. The design of neural network was inspired from functionality of biological nervous system. Neural networks mimics the processing of information as that of human brain. Artificial neural networks are such a novel approach of neural network. An artificial neural network (ANN) is an efficient information processing system which mimics the characteristics of a biological neuron. It consists of a large number of highly interconnected processing elements called nodes. These nodes commonly operate in parallel and are configured in regular architectures. Each node is connected with the other nodes, not necessary to all , by a inter connecting edge which is associated with some weights for the processing of information regarding the input signal. Each node has its own internal state called activation state which is responsible for activating or deactivating the signal and this activated signal is transmitted to other interconnecting nodes. Each node can sent activated signal one at a time. Artificial neural network has three main components.

1. Input layer
2. Hidden layer
3. Output layer

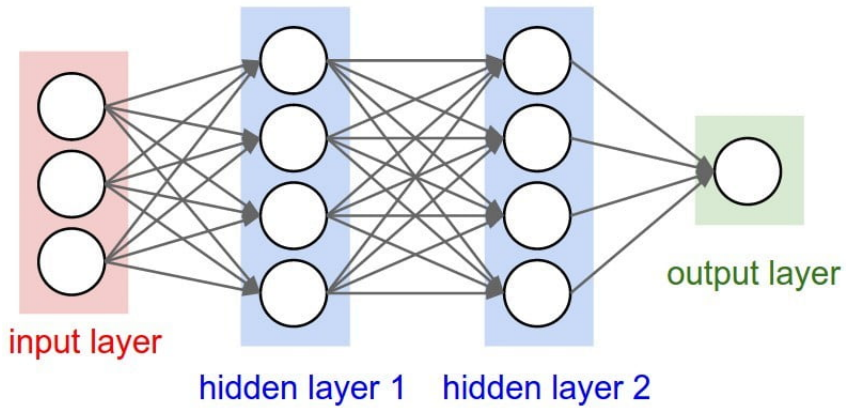

FIGURE 2.3: Model of a typical artificial neural network

Figure 2.3 describe a typical artificial neural network. The inputs are fed to the input layer which is transmitted to the connecting nodes hidden layer (first hidden layer) where the processing is done based on certain criteria and again the output at this layer is fed to the next layer (hidden layer 2) and similar type of processing are done. This process continues till the output is obtained at output layer. Following are six different of ANNs are currently being used.

- Feed forward Neural Networks
- Radial basis function Neural Networks
- Kohonen Self Organising Neural Networks
- Recurrent Neural Networks
- Convolution Neural Networks
- Modular Neural Networks

Fuzzy logic is an approach of multi valued logic based on degree of truth rather than binary truth value of "0" or "1". It is a commonly used mathematical tool that deals with the uncertainty and partial truth. The idea of fuzzy logic was introduced by Lotfi A. Zadeh in 1965. Fuzzy logic provides an important concept of computing to soft computing. It provides a technique to deal with imprecision and information granularity. The fuzzy theory provides a mechanism for representing linguistic concepts such as "high", "low", "medium", "tall", "many" etc. In fuzzy system the values are given in the range from 0 to 1 where 0.0 represents "falseness" and 1.0 represents "absolute truth". Fuzzy set represent fuzzy logic which offers to model the uncertainty and vagueness of the problem or system. The definition of fuzzy set is given below:

- **Definition 1.** Let U be a non-empty set and A fuzzy set in U is characterized by its membership function.

$$\mu_A : U \to [0,1] \tag{2.1}$$

and $\mu_A(u)$ *is as degree of membership of element* x *in fuzzy set* $A \; \forall \; x \in U$
The representation of fuzzy set A is given in eq. 2.2

$$A = \{(u, \mu_A(u)) | u \in U\} \tag{2.2}$$

- If $U = \{u_1, u_2, ..., u_n\}$ is a finite set and A is a fuzzy set in U then the fuzzy set is represented as following:

$$A = \mu_1 / u_1 + \mu_1 / u_1 + ... + \mu_n / u_n \tag{2.3}$$

For example, in real life we commonly say that "*XYZ is a tall person*" can be translated as XYZ belongs to a set of *tall* people and can be represented symbolically as $\mu(tall)$, where μ is the membership function that can give a value between 0 and 1 based on the membership degree. In Asian countries over 150 cm long person may belongs to a tall and in European counties over 180 cm long person belongs to a tall class. So in crisp set theory, the generalization of such cases are hard because there is not a fixed decision point to denote a person may belong to a tall class or not. Therefore, using fuzzy set theory we can handle such ambiguity. In figure 2.4, we have shown the fuzzy membership for the objective term "tall" has been assigned fuzzy values. At 150 cm and below, a person does nor belong to the fuzzy class of tall while for above 180, it totally belongs to a tall class. Between 150 cm to 180 cm we have given fuzzy values in the range of [0,1].

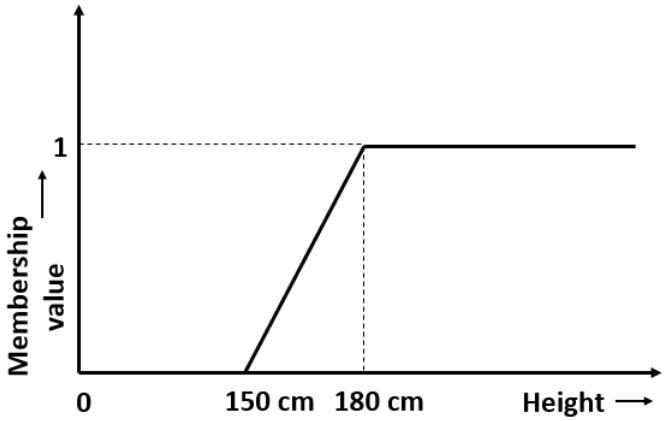

FIGURE 2.4: A fuzzy representation of tall person

Apart from fuzzy set theory, fuzzy logic is used in solving approximate reasoning represented in the form of fuzzy inference engine or fuzzy rule base. The birth of Evolutionary Algorithms arises due the the great success of Genetic Algorithms(GAs). The algorithms which uses the principle of natural evolution are said to be evolutionary algorithms (EAs). There are several algorithms comes under evolutionary algorithms in which Genetic Algorithms (GAs), Particle Swarm Optimization (PSO).

Genetic Algorithms:

Genetic algorithms are the random search and optimization algorithm that is inspired from the process of natural evolution. It works on a set of population of individuals. These individuals are represented by encoding of gene like binary encoding, octal encoding, tree encoding, permutation based encoding etc. The individuals of the population have to pass through the process of genetic evolution (selection,crossover and mutation) and elitist strategy may be used to keep the best or some set of best individuals. This process is continuously repeated and the process terminates if it satisfy the terminating criteria. The terminating criteria can be the given maximum generation or there is no change in elitist individual for a some generation. The detail description of GA is given in Chapter 3

Particle swarm optimization (PSO) is also a evolutionary algorithm which is based on population developed by Dr. Eberhart and Dr. Kennedy in 1995. It is a stochastic optimization technique inspired by social behavior of bird flocking or fish schooling. PSO has many similarities with evolutionary computation techniques such as Genetic Algorithms (GA) that involves a set of random solutions called population and searches for global optima. However, PSO is dissimilar from GA in the sense it does not has evolution operators such as crossover and mutation. In PSO, the solutions are called particles that move through the problem space by following the current optimum particles. Each particle in multidimensional space is associated with a position and a velocity. The main feature of PSO is memory. Each particle has a memory of its best position and knowledge of the swarm's best. Members of a swarm communicate through their memory knowledge and modify their position and velocity. This can be done in two main ways:

1. Global best: It is the best position among the swarm which is known to all. If any best position found then there is immediate impact to update the new best position.

2. Local best: This is best solution within a subset of swarn where each particle only immediately communicates about its best position.

Operation on PSO

Each particle consists of following two component:

- Position vector.....$x_i(t)$
- Velocity vector.....$v_i(t)$

After finding the two best values, the particle updates its velocity and positions with following equation 2.4 and 2.5.

$$v_i(t+1) = v_i(t) + c1 \star rand() \star (pbest(t) - x_i(t)) + c2 \star rand() \star (gbest(t) - x_i(t)) \quad (2.4)$$

$$x_i(t) = x_i(t) + v_i(t) \quad (2.5)$$

During the process, each particle has its own memory having individual knowledge pbest, i.e., its own best-so-far in the position and social knowledge gbest i.e., pbest of its best neighbors Performing the velocity update, using the formula given below above. Here, c1 and c2, the cognition and social components respectively are the acceleration constants which changes the velocity of a particle towards the pbest and gbest, rand is a random number between 0 and 1. The flowchart of PSO is given in figure 2.5

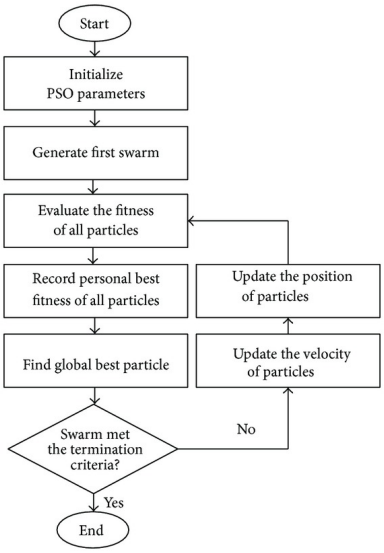

FIGURE 2.5: PSO Flowchat process

2.2.2 Machine Learning

Machine learning is the part of artificial intelligence (AI) that makes machine or system capable to learn automatically from experience in order to improve its performance without explicitly programmed. There are four types of machine learning.

- **Supervised Learning** : As the name depicts the presence of a supervisor or a teacher. Supervised learning is a type of learning in which system learns from a training data set. A training data set consists a number of examples which has both input variables as well as output variables. In other words the output of the given input is already known. The machine or system has been trained on such data set. The goal is to approximately map the input variables to output variable so that when a new input will be given then the machine can predict the output. This can be explained with a real life example. Before the examination, the students are first taught by teachers regarding some topics

or subjects to develop their skills. Some new questions are made regarding the same topic they have been taught. Still some of the students are capable enough to solve the given questions. Such learning process are called supervised learning.

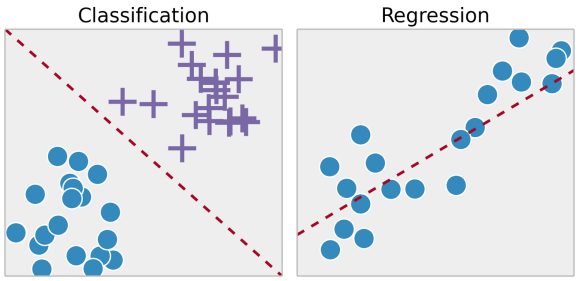

FIGURE 2.6: Examples of Supervised Learning. Available from: https://www.bing.com/images/search?q=supervised+learning+image&FORM=HDRSC2

Key Points:

- Most of the problem are based on Regression and classification problem.
- The training data are labeled.
- There are some popular algorithms like Linear Regression, SVM(support vector machine), Random Forest, ANN (artificial neural network), Decision Tree, Naive Bayes, Nearest Neighbor.
- It is generally used for Predicting Modeling.

- **Unsupervised Learning** : Unsupervised Learning is another type of machine learning in which there is no need any supervision. It learns from the data set having only input data and no corresponding output. The goal of unsupervised learning is to find the hidden pattern or underlying structure in a way to learn more about data. For example, if a kid is given a basket of fruits having apples and bananas. The kid starts to differentiating the apples and bananas based on his own intuition of color, shape and size. There is no one there to guide him. Such type of learning are unsupervised leaning.

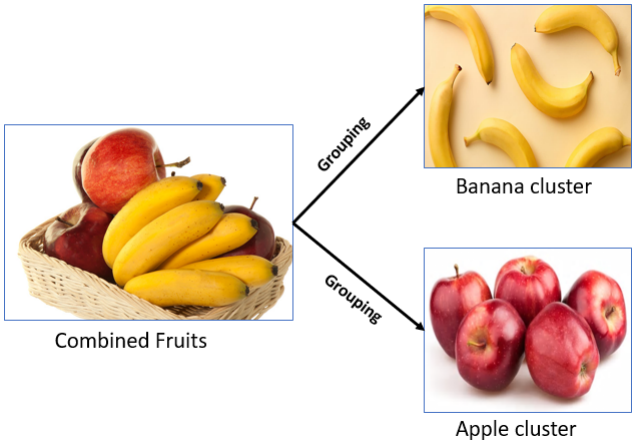

FIGURE 2.7: Example of an Unupervised Learning

Key Points:

- It is widely used for clustering problem and anomaly detection.
- The dataset are unlabeled.
- Some most popular algorithms are K-means clustering, FCM, DB-Scan, Hierarchical clustering, SOFM clustering etc.
- It is generally used for Descriptive Modeling.

- **Semi-supervised Learning** :This learning process lies between supervised and unsupervised learning because it uses both labeled and unlabeled data. During training phase there are small amount of labeled data and large amount of unlabeled data. The process of labeling large amount of data for supervised learning is time consuming as well as expensive. The labeling of too much data may result in biasing the model. So including large amount of unlabeled data may improve improve the accuracy of model which results in time saving and cost efficient.
Key Points:

- **Reinforced Learning** : This is another type of learning process which is exposed to an environment where it is trained on the basis of trail and error method. In this learning , an agent interacts with an environment to do some action based on its previous history to complete each step. For every successful step it gets reward and for failure it will be punished.

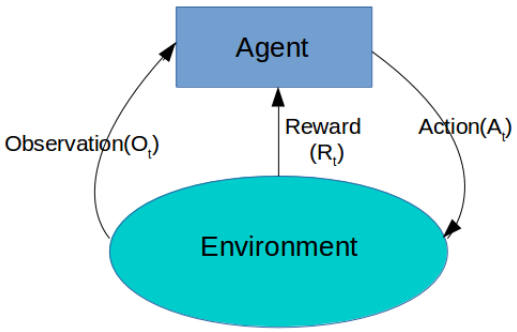

FIGURE 2.8: Examples of Reinforced Learning

Key Points:

- Markov Decision Process is basic model for reinforcement learning .
- The most popular algorithms are Q-Learning, Deep Adversarial Networks.
- Its practical application in self driving cars, games like chess.

2.2.3 Statistics

Statistics is one of the important part to handle data mining tasks. It provide tools and methods to give deeper insight into data. Statistical knowledge helps to select appropriate methods to collect data, employ the proper analysis and efficiently present the obtained information. It is rvery important in decision making and prediction based on data. There are some most important statistical distribution needed to analyze data.

TABLE 2.1: An information Table

Individual number	*Headache*	*Malaria*	*Temperature*	*Flu*
x_1	Yes	Yes	Nominal	No
x_2	Yes	Yes	High	Yes
x_3	Yes	Yes	Very High	Yes
x_4	No	Yes	Nominal	No
x_5	No	No	Highl	No
x_6	No	Yes	Very High	Yes

- *Poisson Distribution*: A Poisson distribution is a statistical distribution that shows how likely the number of times an event occurs within a specific period of time. It is used for an independent event which occurs at some constant rate within a specific time interval.
- *Binomial Distribution*: It is a probability distribution of one of the two outcomes of a random event has occurred a fixed number of trails.
- *Hyper geometric Distribution*: The hyper geometric distribution is is discrete probability distribution used to calculate probabilities when sampling without replacement.
- *Discreate Uniform Distribution*: It is a type of uniform distribution which is discrete and symmetric.
- *Negative Binomial Distribution* It is a probability distribution performed for discrete random variable to get specific number of failure on a series of independent and identical distributed bernoulli trails.

Apart from statistical distribution, a number of statistical models like Bayesian model (Naive Bayes), Markov model, Gaussian mixture model, Hidden Markov Random Field model etc. commonly used to validate the model taken under analysis. These models are widely used in machine learning.

2.2.4 Roughset Theory

The advancement of computer science and technology in the field of computer network, a huge amount of information are processed every second of the day. The data needed for processing may be consistent or inconsistent. There are already developed like probability theory, fuzzy set theory and evidence theory to handle uncertainty. Rough set theory is a new paradigm for dealing with vague, imprecise, inconsistent and uncertain knowledge. Because of its unique approach and easy operation, the rough set theory has become an important mathematical tool in the field of intelligent information processing [78],[79]. In recent years it become an important tool for data mining and used in knowledge discovery, decision support and analysis, machine learning.

A knowledge representation scheme can be defined formally by an information system S expressed as the 4-tuple.

$$S = \langle U, R, V, f \rangle, R = C \cup D \tag{2.6}$$

Where U is a finite nonempty set of objects and R is a finite nonempty attributes formed by union of the subsets C and D are called condition attribute set and decision attribute set respectively. V is a finite set of values formed by union of the set of values of attribute a, V_a and card V_a>1. f is a description or information function. In Table 2.1, the set $U = \{x_1, x_2, x_3, x_4, x_5, x_6\}$ is a finite nonempty set, also called a universe, and $R =$

$\{Headache, Malaria, Temperature, Flu\}$ is a finite nonempty set, also called an attribute set. Some basic concepts [80] on rough set theory are given below.

- Indiscernible relation: Given a subset of attribute set $B \subseteq R$, a relation is said to an indiscernible relation if it satisfy the following relation.

$$ind(B) = \{(x,y)|(x,y) \in U^2, \forall_{b \in B}(b(x) = b(y))\} \tag{2.7}$$

  The equivalence relation is an indiscernible relation and the equivalence class is denoted by $[x]_{ind(B}$, or$[x]_B$, or$[x]$ where x is an object.

- Upper and lower approximation sets: The upper and lower approximation sets are defined on a subset, $X \subseteq U$, on 2.8 and 2.9 respectively.

$$\overline{apr}(X) = x \in U|[x] \cap X \neq \phi \tag{2.8}$$

$$\underline{apr}(X) = x \in U|[x] \subseteq X \tag{2.9}$$

  If an object $x \in POS(X)$, then it belongs to target set X certainly. If an object $x \in BND(X)$, then it doesn't belong to target set X certainly. If an object $x \in NEG(X)$, then it cannot be determined whether the object x belongs to target set X or not. Where the $POS(X) = \underline{apr}, BND(X) = \overline{apr}(X) - \underline{apr}(X)$ and $NEG(X) = U - \overline{apr}$.

- Definable sets: For a given an information system,a target set ($X \subseteq U$) is definable with respect to attribute subset ($B \subseteq R$) if $\overline{apr}(X) = \underline{apr}(X)$.

- Rough Sets: For a given an information system,a target set ($X \subseteq U$) is definable with respect to attribute subset ($B \subseteq R$) if $\overline{apr}(X) \neq \underline{apr}(X)$.

- Roughness of rough set: The roughness of rough set of a target set $X(X \subseteq U)$ with respect to attribute set $B(B \subseteq R)$ is defined as follows:

$$P_B(X) = 1 - \frac{|\underline{apr}(X)|}{|\overline{apr}(X)|} \tag{2.10}$$

  where, $X \neq \phi$ otherwise $P_B(X) = 0$; $||$ denotes the cardinality of a finite set .

There are some primitive operations on rough sets [81]

- Union operation: $\overline{apr}(X \cup Y) = \overline{apr}(X) \cup \overline{apr}(Y)$ and, $\underline{apr}(X \cup Y) \supseteq \underline{apr}(X) \cup \underline{apr}(Y)$
- Intersection operation:$\overline{apr}(X \cap Y) \subseteq \overline{apr}(X) \cap \overline{apr}(Y)$ and, $\underline{apr}(X \cap Y) \subseteq \underline{apr}(X) \cap \underline{apr}(Y)$
- Difference operation: $\underline{apr}(X - Y) = \underline{apr}(X) - \overline{apr}(Y)$ and, $\overline{apr}(X - Y) \supseteq \overline{apr}(X) \cup \underline{apr}(Y)$
- Complementary operation:$\sim \overline{apr}(X) = (\sim X))$ and, $\sim \underline{apr}(X) = apr(\sim X))$ where $\sim X = U - X$.

2.2.5 Data Visualization

Data visualization is also one the important tools for data mining. It is a generic term widely used to represent data or information in the form visual. It is difficult to understand digits or unknown texts to interpret. This is where data visualization comes into play. This makes data mining task much easier to describe or analyze the behavior of of textual or numerical data.

Data visualization discloses some hidden trends and information which may go unnoticed. Apart from saving time, there is much more importance of data visualization in the process of decision making.

There are different methods of data visualization methods exist. The visualization is done based on data. Data can be univariate, bivariate and multivariate.

- Univariate: The measurement is done only for single quantitative variable and characterized by distribution. This includes visualization methods like histogram, bar plot, pie chart etc.
- Bivariate: It measures by constituting the pair of two related variables. The common visualization method includes line graphs, scatter plots etc.
- Multivariate: It the the multidimensional representation of a variable. The common method used for visualizations are pixel based method, icon based method, dynamic parallel coordinate system.

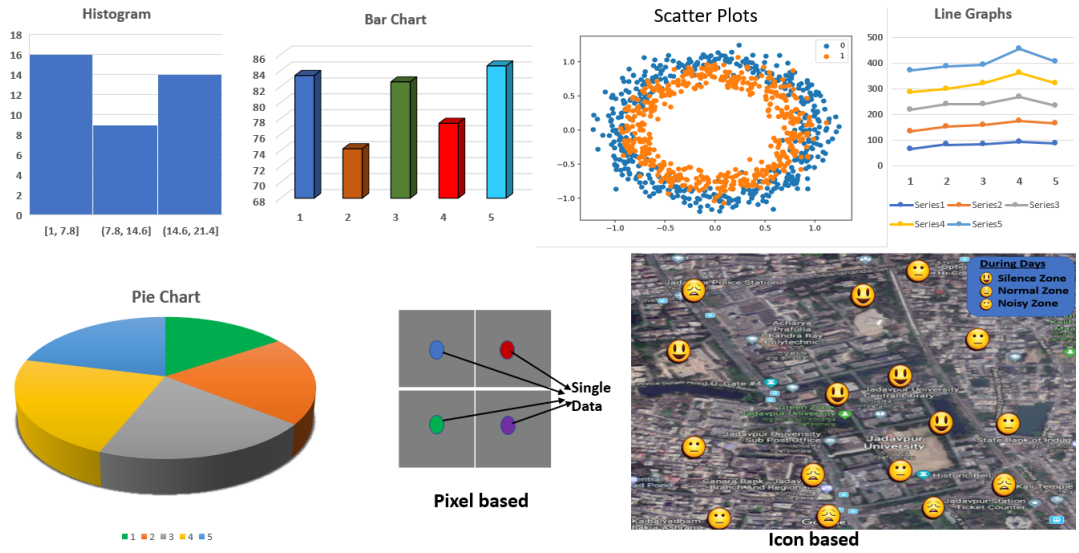

FIGURE 2.9: Examples of some common visualization methods

Figure 2.9 shows some common visualization method method.

2.3 Applications of Data Mining

These are some popular applications of data mining which are as follows:

- **Future Healthcare:** Data mining holds great potential to improve health systems. It uses data and analytics to identify best practices that improve care and reduce costs. Researchers use data mining approaches like multi-dimensional databases, machine learning, soft computing, data visualization and statistics. Mining can be used to predict the volume of patients in every category. Processes are developed that make sure that the patients receive appropriate care at the right place and at the right time. Data mining can also help healthcare insurers to detect fraud and abuse.

- **Market Basket Analysis:** Market basket analysis is a modelling technique based upon a theory that if you buy a certain group of items you are more likely to buy another group of items. This technique may allow the retailer to understand the purchase behavior of a buyer. This information may help the retailer to know the buyer's needs and change the store's layout accordingly. Using differential analysis comparison of results between different stores, between customers in different demographic groups can be done.

- **Education:** There is a new emerging field, called Educational Data Mining, concerns with developing methods that discover knowledge from data originating from educational Environments. The goals of EDM are identified as predicting students' future learning behavior, studying the effects of educational support, and advancing scientific knowledge about learning. Data mining can be used by an institution to take accurate decisions and also to predict the results of the student. With the results the institution can focus on what to teach and how to teach. Learning pattern of the students can be captured and used to develop techniques to teach them.

- **Manufacturing Engineering:** Knowledge is the best asset a manufacturing enterprise would possess. Data mining tools can be very useful to discover patterns in complex manufacturing process. Data mining can be used in system-level designing to extract the relationships between product architecture, product portfolio, and customer needs data. It can also be used to predict the product development span time, cost, and dependencies among other tasks.

- **CRM:** Customer Relationship Management is all about acquiring and retaining customers, also improving customers' loyalty and implementing customer focused strategies. To maintain a proper relationship with a customer a business need to collect data and analyze the information. This is where data mining plays its part. With data mining technologies the collected data can be used for analysis. Instead of being confused where to focus to retain customer, the seekers for the solution get filtered results.

- **Fraud Detection:** Billions of dollars have been lost to the action of frauds. Traditional methods of fraud detection are time consuming and complex. Data mining aids in providing meaningful patterns and turning data into information. Any information that is valid and useful is knowledge. A perfect fraud detection system should protect information of all the users. A supervised method includes collection of sample records. These records are classified fraudulent or non-fraudulent. A model is built using this data and the algorithm is made to identify whether the record is fraudulent or not.

- **Intrusion Detection:** Any action that will compromise the integrity and confidentiality of a resource is an intrusion. The defensive measures to avoid an intrusion includes user authentication, avoid programming errors, and information protection. Data mining can help improve intrusion detection by adding a level of focus to anomaly detection. It helps an analyst to distinguish an activity from common everyday network activity. Data mining also helps extract data which is more relevant to the problem.

- **Lie Detection:** Apprehending a criminal is easy whereas bringing out the truth from him is difficult. Law enforcement can use mining techniques to investigate crimes, monitor communication of suspected terrorists. This filed includes text mining also. This process seeks to find meaningful patterns in data which is usually unstructured text. The data sample collected from previous investigations are compared and a model for lie detection is created. With this model processes can be created according to the necessity.

- **Customer Segmentation:** Traditional market research may help us to segment customers but data mining goes in deep and increases market effectiveness. Data mining aids in aligning the customers into a distinct segment and can tailor the needs according to the customers. Market is always about retaining the customers. Data mining allows to find a segment of customers based on vulnerability and the business could offer them with special offers and enhance satisfaction.

- **Financial Banking:** With computerized banking everywhere huge amount of data is supposed to be generated with new transactions. Data mining can contribute to solving business problems in banking and finance by finding patterns, causalities, and correlations in business information and market prices that are not immediately apparent to managers because the volume data is too large or is generated too quickly to screen by experts. The managers may find these information for better segmenting,targeting, acquiring, retaining and maintaining a profitable customer.

- **Corporate Surveillance:** Corporate surveillance is the monitoring of a person or group's behavior by a corporation. The data collected is most often used for marketing purposes or sold to other corporations, but is also regularly shared with government agencies. It can be used by the business to tailor their products desirable by their customers. The data can be used for direct marketing purposes, such as the targeted advertisements on Google and Yahoo, where ads are targeted to the user of the search engine by analyzing their search history and emails.

- **Research Analysis:** History shows that we have witnessed revolutionary changes in research. Data mining is helpful in data cleaning, data pre-processing and integration of databases. The researchers can find any similar data from the database that might bring any change in the research. Identification of any co-occurring sequences and the correlation between any activities can be known. Data visualisation and visual data mining provide us with a clear view of the data.

- **Criminal Investigation:** Criminology is a process that aims to identify crime characteristics. Actually crime analysis includes exploring and detecting crimes and their relationships with criminals. The high volume of crime datasets and also the complexity of relationships between these kinds of data have made criminology an appropriate field for applying data mining techniques. Text based crime reports can be converted into word processing files. These information can be used to perform crime matching process.

- **Bio Informatics:** Data Mining approaches seem ideally suited for Bioinformatics, since it is data-rich. Mining biological data helps to extract useful knowledge from massive datasets gathered in biology, and in other related life sciences areas such as medicine and neuroscience. Applications of data mining to bioinformatics include gene finding, protein function inference, disease diagnosis, disease prognosis, disease treatment optimization, protein and gene interaction network reconstruction, data cleansing, and protein sub-cellular location prediction.

Apart from these applications, community detection is emerged as one of the prime application of data mining due to the advancement of modern network science.

Chapter 3

Genetic Algorithms

3.1 Introduction to Genetic Algorithms

Charles Darwin was an English naturalist who published his book "On the origin of Species" in 1859. In this book he proposed the theory of natural evolution, in the process of natural evolution, the organisms change over time. As a result, the change in inheritable character allows organisms to adapt to the environment to survive and produce their offspring. The theory sometimes called "survival of the fittest".

The genetic algorithm is also inspired by such biological natural evolution. It copies the process of natural evolution that involves changes like selection, crossover, and mutation. Genetic algorithms are the random search and optimization algorithm. Figure 3.1 shows the procedure of GA. It starts with a set of a population of individuals who are randomly initialized. These individuals are represented by the encoding of a gene like binary encoding, octal encoding, tree encoding, permutation-based encoding, etc. A fitness evaluation is done on the population and then the individuals of the population have to pass through the process of genetic evolution (selection, crossover, and mutation) and elitist strategy may be used to keep the best or some set of best individuals. Alternatively, we can use any replacement policy. This process is continuously repeated and the process terminates if it satisfies the terminating criteria. The terminating criteria can be the given maximum generation or there is no change in elitist individual for some generation.

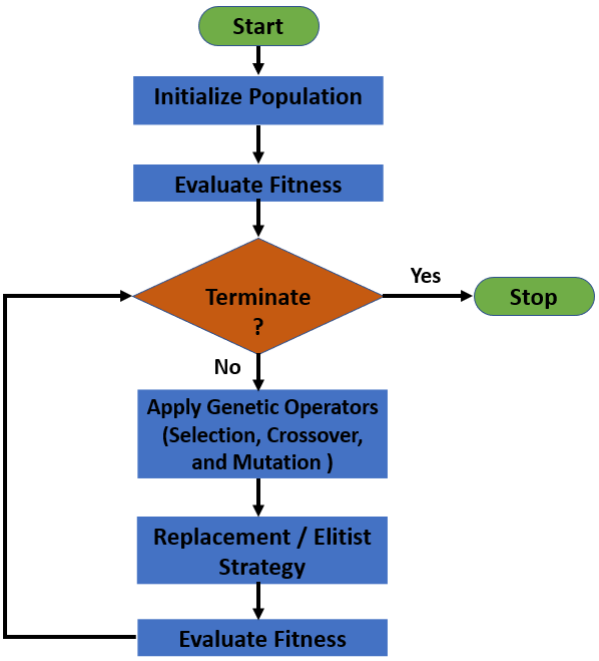

FIGURE 3.1: A flowchat of Genetic Algorithm

3.2 Terminologies in Genetic Algorithm

There are some important key terminologies related to genetic algorithm (GA). These terminologies are discussed below.

3.2.1 Individuals

An individual is a single solution in GA. This solution can be represented in two form named genotype and phenotype.

- In genotype, each chromosome is the raw (genetic) information which the GA deals.
- The phenotype is the expressive representation of a chromosome in the terms of the model.

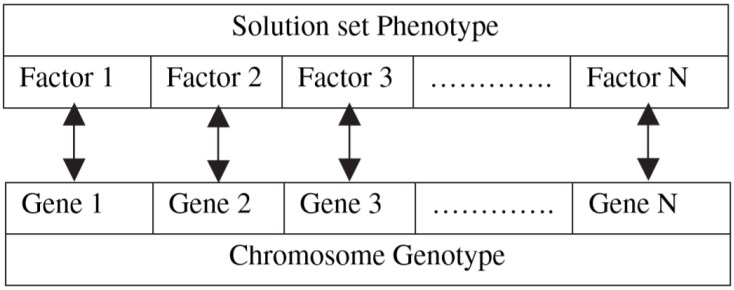

FIGURE 3.2: Representation of Genotype and phenotype

A chromosome is a combination of genes.

3.2.2 Genes

Genes are the building block of a Generic Algorithms. A chromosome is formed by sequencing the genes. A Gene is not the solution to a problem but it may describe a possible solution. Each gene in a chromosome is a controlling factor that has some upper and lower bound. The representation of the gene is shown in figure 3.3 which consists of four binary strings. Each binary string of a gene can take a value of either 0 or 1.

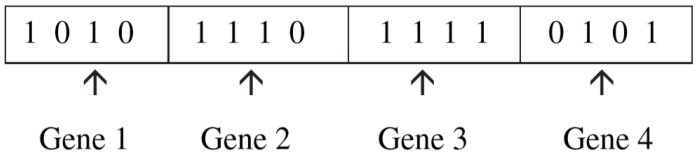

FIGURE 3.3: Representation of Genes

3.2.3 Fitness

In the Genetic algorithm, the fitness is a value of a chromosome or individual obtained from the objective function taken into consideration. The chromosome is first decoded and then the objective function will be calculated. The fitness is the measure of goodness of an individual. Higher the fitness value, closer will the chromosome to the optimal solution.

3.2.4 Populations

A population is a set that consists of a collection of individuals. The individuals in a population have to pass through different stages of evolution. There are two main aspects of the population that are used in Genetic Algorithms.

1. The initial population generation: Before the start of the GAs process. Usually, the finite size of the initial population commonly called population size is created randomly from different chromosomes. Some times a heuristic approach may be used to generate an initial population to obtain some quality of solutions.

2. The population size: It is the number of individuals required to generate the population initially. The population size is a very important part of GAs. The size of the population depends upon the problems. If the size of the population is high then it is to explore the search space. The time complexity of GA is $O(nlogn)$ where n is the population size. But for higher population size, it will be the cost expensive. It is the duty of the programmer to select the appropriate population size.

3.2.5 Encoding

Encoding is the method to design the gene of an individual. The process of representing each gene may be performed using bits, real number, trees, arrays or any other things. The encoding mainly depends upon the nature of the problem. There are different types of the commonly used encoding scheme for representing a chromosome.

- Binary encoding: In this encoding strategy, each gene of a chromosome is encoded a binary string (i.e 0 and 1) which can represent some characteristics of the solution. There is another possibility that the whole chromosome may represent a number.

- Octal encoding: In this encoding scheme the strings are represented using an octal number between 0-7.

- Hexadecimal encoding: In this encoding scheme the strings are represented using a hexadecimal number between 0-9 and A-F.

- Real Number encoding: The gene of chromosome in real encoding is an integer. The sequence of the strings constructs a chromosome. It is sometimes called permutation encoding. Permutation encoding is helpful for ordering problems.

- Value Encoding: In value encoding, each gene of a chromosome is a string of some values. The values can be integers, real numbers, chars, some complicated object or anything appropriate to the problem. This type of encoding can be used in the problems which are concerned with some complicated values like real numbers, complex number, etc. are used. It is very difficult to use binary encoding for this type of problem.

- Tree encoding: This encoding scheme is mainly used in genetic programming to evolve some expression of a program. Functions and commands of a programming language can be treated as a tree object and each chromosome represents a tree object.

3.2.6 Genetic Operators

The genetic operators are the heart of the genetic algorithm. During the process of evolution, a chromosome has gone through different stages of genetic operators and create offspring(s). The genetic operator consists of selection, crossover and mutation operations.

Selection

Selection is the process in which parents are randomly chosen from the population for crossover. In many problems, the selection is used to create a mating pool based on objective or fitness function and then parents from created mating pool go for crossover. The chromosome having higher fitness value has a higher chance to be selected (i.e higher will be selection pressure the better chromosomes are favored). The selection pressure is the degree to which the fitter individuals are favored. This selection pressure controls the GA to improve the fitness of the population over the upcoming generations. Selection pressure has to be balanced with crossover and mutation. Too high selection pressure leads to a high fit chromosome carry over the population which results in a reduction in population diversity. Too low selection pressure will lead to a very slow evolution. The following are the different selection methods that are commonly used.

- Roulette Wheel Selection: It is the traditional selection strategy of GA. It is based on a proportionate selection technique where a chromosome is selected from the mating pool with a probability proportional to its fitness. In a Roulette wheel, the slots in the wheel are given weighted proportion to the individual's fitness values. The weighted proportion is calculated as follows: Roulette wheel selection is simple and easy to implement but it is noisy. The rate of evolution depends on the fitness variance.

- Random Selection: This selection method randomly selects a parent from the population. But it has a higher capability to disrupt the genetic code than the Roulette Wheel.

- Rank Selection: The Roulette wheel selection strategy will suffer when there is a large difference in fitness. If the fitness of the best solution is around 90 percent, its circumference occupies 90 percent of the Roulette wheel, then other solutions have very few chances to be selected. It selects the chromosomes based on ranking. The best chromosome has the highest fitness (i.e N) and the worst chromosome has given the least fitness(i.e 1).

- Tournament Selection: It is the selection strategy that balances the selection pressure. The size of the tournament determines the selection pressure. A set of chromosomes having size equals to tournament size is randomly chosen from the population. The best individual from the tournament is the winner having the highest fitness will form a mating pool. It is the responsibility of the programmer to decide the appropriate tournament size.

- Boltzmann Selection: In Boltzmann selection, a varying temperature decides the rate of selection. Initially, the temperature is given high which means low selection pressure. The temperature is falling gradually, which results in an increase in selection pressure, so this allows the GA to select solutions from the best part of the search space while maintaining the appropriate diversity in the population. The equation of Boltzmann probability is given in 3.1.

$$p = \frac{exp - (f_{max} - f(X_i))}{T} \tag{3.1}$$

  Where, $T = T_0(1-\alpha)^k$ and $k = (1 + 100 * g/G), g$ is the value of generation number and G is the maximum value of generation. The value of α lies in the range of [0, 1] and T_0 range from [5,100]. The optimal solution is achieved when the value of T reaches zero.

- Elitism: It is another type of selection strategy which reserves the best or some set of best chromosomes found so far to the new population. This strategy is very useful if

the best chromosome may get lost due to crossover and mutation or maybe in selection. This will improve the efficiency of GAs.

3.2.7 Crossover

Crossover is the process in which two parent solutions recombine and produce a child. The crossover operation is applied to the chromosomes of the mating pool after the selection process. The selection process does not generate any new chromosome, it only enriches the population with better chromosomes. But the crossover operator may generate a new child different from the chromosomes in the population.

- One Point Crossover: It is a traditional crossover operator of GA which is commonly used. The figure of one point crossover is shown in Figure 3.4(a). In this crossover operation, a pair of random chromosomes are selected from the mating pool and then a crossover site is also randomly selected along the length of the chromosome. Cut both parents at the crossover point and the combine the left cut part of the first parent with the right cut part of the second parent and similarly left cut part of second parent is merged with the right cut part of the first parent. If an appropriate crossover site and good parents are selected then a better child will be produced otherwise it may hamper the quality of the generated child.

- Two Point Crossover: Two point crossover is similar to one point crossover, but the difference is two crossover sites. Addition of additional crossover point may result in more exploitation of search space. But it may disrupt the building block of GA which may result in a decrease in performance. The pictorial representation of two point crossover is given in Figure 3.4(b)

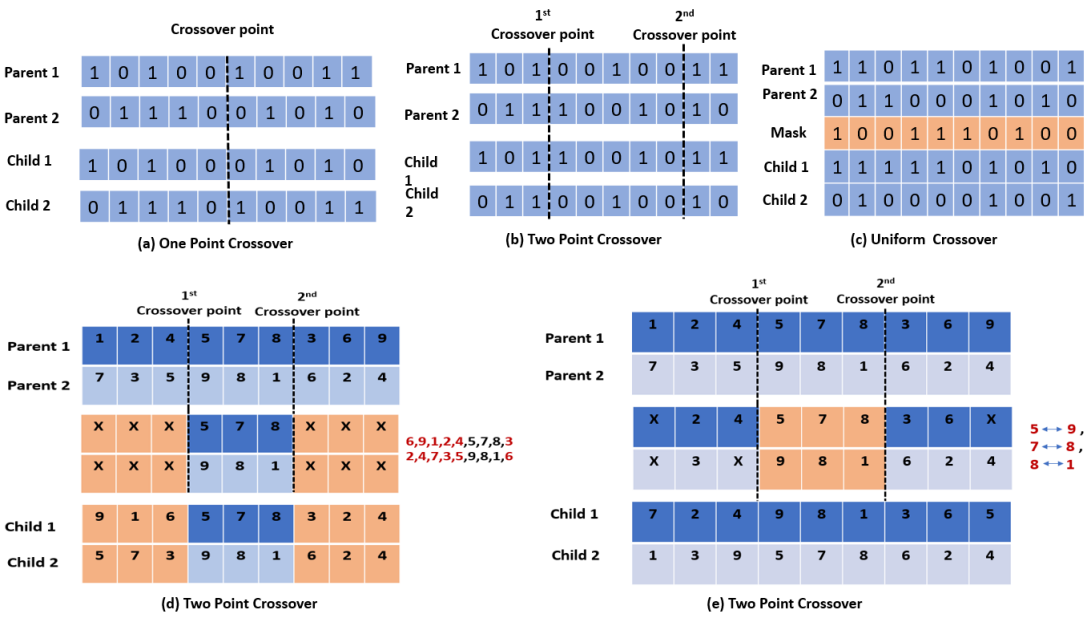

FIGURE 3.4: Different Crossover Operations

- Multipoint Crossover: There are generally two possible of multipoint crossover or N point crossover. The first one is an even number of crossover sites and the second one is an odd number of crossover sites. In the first case, the number of crossover sites is randomly chosen around a ring and the information is exchanged. In the case of an odd number of crossover sites, a different crossover point is selected at the beginning.

- Uniform Crossover: Uniform crossover is different from N point crossover. In this crossover, a binary mask of length equal to the length of the chromosome is generated. For the bit "0" in the mask, the child inherits the information from one parent and for bit "1" it inherits from the second parent. Figure 3.4(c) shows the process of this crossover.

- Shuffle Crossover: It is quite related to uniform crossover. At first, a single crossover point is selected. But before the exchange of information, both parents are randomly shuffled in the same manner. After recombination, the genes in both offspring are unshuffled. This avoids positional biasing.

- Ordered Crossover: In this crossover, two random crossover points are selected which partition the parents into the left, middle and right portion. the first child inherits the left and right part from the first parent and the middle part is determined by the genes of the middle portion of the first parent with the order in which the values appeared in the second parent. The process of this method can be represented in Figure 3.4(d).

- Partially Matched Crossover: In this crossover, both parents are aligned as strings of genes and two crossover points are randomly selected along the strings of the chromosome. The middle portion has a matching section that is transferred from one parent to another. It proceeds by position wise exchanges. Figure 3.4(e) illustrates these exchanges.

3.2.8 Mutation

- Flipping: In this mutation, it involves changing of bit 0 to bit 1 and vice-versa.

- Interchanging: This mutation process involves selection of two random positions of the strings and the bits corresponding to those positions are interchanged.

- Reversing: In this mutation, a mutation position is randomly chosen and the bits next to selected mutation position are reversed and the child chromosome is produced.

3.3 Replacement

Replacement is the last stage of any genetic process. Two parents are drawn from a specified size of a population. they go through the genetic operations to produce two children. Since the size of the population is fixed so all the parents and children cannot be fitted in a fixed size of the population. So to make the new population of the same size as it was previous, some parents, children or both should be removed.

- Random Replacement: In this replacement scheme, two individuals are chosen randomly from the population is replaced by the children. The parent chromosome can also be the candidates for selection.

- Weak Parent Replacement: This is also a replacement policy in which a weak parent is replaced by a strong child.

3.4 Terminating condition

- Maximum generations: It the specified number of generation or iteration that the user has decided to stop the genetic algorithm.

- Elapsed time: It is the specified time duration that the GA has spent to stop. If the GA has reached the specified maximum number of the generation before the specified time has elapsed, the process will stop based on the maximum generation.
- No change in fitness: The genetic algorithm will stop if there is no change in fitness of the best individual found so far for a specified number of generations. Note: GA will stop if it satisfies the maximum number of the generation before the specified number of a generation with no change.
- Stall generations: The genetic process will stop if there is no improvement in the objective function for a series of consecutive generations of length Stall generations.
- Stall time limit: The genetic process will stop if there is no improvement in the objective function during an interval of time in seconds equal to the Stall time limit.

There are several advantages of genetic algorithm.

1. It has inbuilt parallelism.
2. GA provides liability because it is capable of solving constrained and unconstrained optimization problems.
3. GA widens the search space.
4. The fitness landscape is complex.
5. It may discover global or near global optimum.
6. It uses only function evaluations.
7. It is simple and easy to modify for different problems.

Some limitation of genetic algorithm includes,

1. It is very difficult to identify the fitness function.
2. It is a hard task to decide the appropriate encoding scheme.
3. Premature convergence may occur.
4. It is very difficult to tune various parameters like the size of the population, mutation rate, cross over rate, the selection method and its strength.
5. It does not use gradients.
6. It does not always provide a guarantee for the optimal solution.

Applications of Genetic Algorithm Genetic algorithms have been used commonly for solving difficult problems (such as NP-hard problems). It is also used in machine learning and also for evolving simple programs. They have been also applicable for some art, for evolving pictures and music. There are some major applications of GA which are as follows:

- Nonlinear dynamical systems predicting, data analysis
- Robot trajectory planning
- Evolving LISP programs(genetic programming)
- Strategy planning

- Finding shape of protein molecules
- TSP and sequence scheduling
- Functions for creating images
- Control gas pipeline, pole balancing, missile evasion, the pursuit
- Design semiconductor layout, aircraft design, keyboard configuration, communication networks
- Scheduling manufacturing, facility scheduling, resource allocation
- Machine Learning Designing neural networks, both architecture, and weights, improving classification algorithms, classier systems
- Signal Processing filter design
- Combinatorial Optimization set covering, traveling salesman (TSP), Sequence scheduling, routing, bin packing, graph coloring, and partitioning

3.5 MOEA

A multi-objective optimization problem is one which has more than one objective functions. These objective functions need to be optimized with certain constraints. The mathematical formulation of multi-objective problem is given below:

$$\begin{aligned} &Maximise/Minimize \;\; f_i(x) \\ &\quad subject\ to \\ &g_j(x) \leq 0 \;\; j = 1,2,...,J \\ &h_k(x) = 0 \;\; k = 1,2,...,K \end{aligned} \tag{3.2}$$

Where x is a vector whose dimension depend upon the number of features taken in hand. f is objective function, g and h are constraint condition.

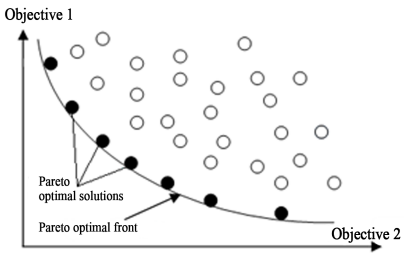

FIGURE 3.5: Dominated and Non-dominated solutions. Available from: https://journals.plos.org/plosone/article/figure?id=10.1371/journal.pone.0154080.g002

The solution of such problem can be expressed through non-dominated solutions or points. In a maximization problem, a solution x_1 (say) is said to be partially less than another solution x_2, ($x_1 \prec x_2$), if there does not exist any value of x_1 greater than x_2. In other word, no value of x_2 is less than x_1 and atleast in any objective the value of x_2 must be greater than x_1. So we say that the solution x_2 dominates solution x_1 or the solution x_1 is inferior to solution x_2 [98]. A solution is said to be non-dominated if it is not dominated by any other

solution. Such solutions is said to be Pareto optimal solutions and the set of Pareto optimal solutions are called Pareto optimal front. Figure 3.5 shows the set of Pareto optimal solution which lie on the curve known as Pareto optimal solutions and the curve called Pareto front while the rest of the solutions are dominated.

3.6 NSGA

There are different version of Genetic algorithms exist in the literature based on multi-objective GA. Non-dominated Sorting Genetic Algorithm [69] is the first multi-objective Evolutionary algorithm based on Genetic Algorithm proposed by Deb which is based non-domination. The flowchart of NSGA is given in figure 3.6. It differs from basic GA in the sense of selection and the crossover and mutation operation remains the same. Before selection, the non-dominated solutions are identified from the population at the current iteration. These non-dominated solutions are given a high dummy fitness value. These solutions share the assigned dummy fitness in order to maintain diversity in the population. The sharing of fitness for each solution can be calculated using its niche count. The niche count of an individual in a currently non-dominated solution is calculated as the sum of its sharing function values for all non-dominated individuals. The shared fitness is computed by dividing the assigned dummy fitness value by niche count.

$$Sharing(d_{ij}) = \begin{cases} (1 - (\frac{d_{ij}}{\sigma_{Share}})^2, & if d_{ij} < \sigma_{Share}; \\ 0, & if d_{ij} > \sigma_{Share}; \end{cases} \tag{3.3}$$

After assigning sharing fitness, temporarily ignore the non-dominated solutions to process the rest of the solutions in the same way. These new non-dominated solutions are then assigned a new dummy fitness value which should be less than the minimum of shared fitness in the previous front. This process is repeated until the whole population is classified into several fronts. The computational time complexity for NSGA-I is $O(MN^3)$ where M is the total number of objective function used and N is the size of the population taken.

3.7 NSGA-II

There are certain limitations in the first NSGA which need to rectify. These limitations are the following:

- High computational complexity of nondominated sorting: The computational complexity of NSGA-I is $O(MN^3)$ where M is the number of objective functions and N is the size of the population. This becomes computationally expensive if the size of the population becomes high.
- Lack of elitism: NSGA-I does have any elitism strategy to keep track of a good solution. Elitism will increase its efficiency.
- Need to specify sharing parameter: There is a sharing parameter "σ_{Share}" which needs to specified by users. This is an additional overhead to accurately specify its value.

So an improved version of NSGA is proposed by K.Deb named NSGA-II [70]. This new version is different from the previous version in non-dominated sorting and diversity maintenance. The non-dominated sorting in NSGA-II has two entity which are

1. Domination count: The number of solutions that dominates the current solution.
2. A set of solution which is inferior to the current solution.

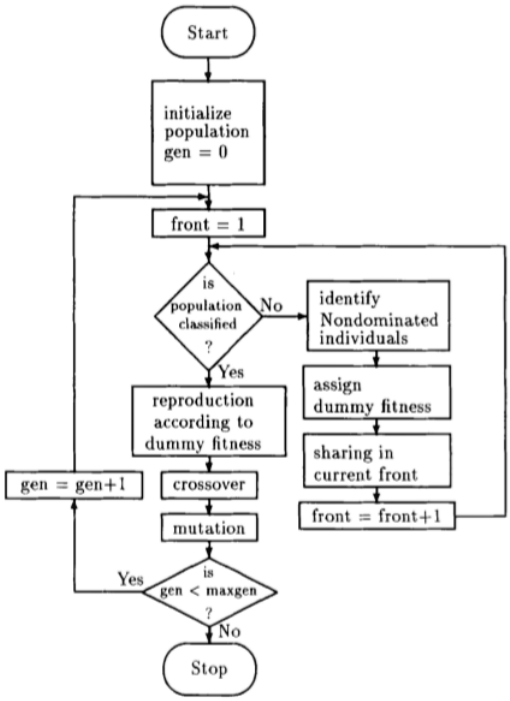

FIGURE 3.6: Flowchart of NSGA-I

These comparisons required to non-dominated sorting. The algorithm for non-dominated sorting of the population is given in

Algorithm 1 Non-dominated-sort(P)

Input: **P** : Population of solution;
Output: F: Pareto Fronts;

for each $p \in P$ **do**
  $S_p = \phi$
  $n_p = 0$
  for each $q \in P$ **do**
    if $p \prec q$ **then**
      $S_p = S_p \bigcup \mathrm{q}$
    else if $q \prec p$ **then**
      $n_p = n_p + 1$
    end if
  end for
end for
if $n_p == 0$ **then**
  $p_{rank} = 1$
  $F = F \bigcup \mathrm{p}$
end if
$i = 1$
while $F \neq \phi$ **do**
  $Q = \phi$
  for each $p \in F_i$ **do**
    for each $q \in S_p$ **do**
      $n_q = n_q - 1$
      if $n_q == 0$ **then**
        $q_{rank} = i + 1$
        $Q = Q \bigcup \mathrm{q}$
      end if
    end for
  end for
  $i = i + 1$
  $F_i = Q$
end while

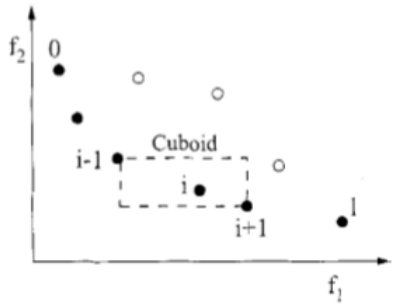

FIGURE 3.7: crowding distance

In order to identify the non-dominated solutions, each solution p in population is assigned the domination count (n_p) and the set of inferior to solution p denoted as S_p. If the domination count n_p equals to zero then these solutions are the non-dominated solutions and it becomes a member of the first front. Now for each solution, p having n_p, visit each element of q of the inferior set S_p and decrease its domination count by one. After doing this, if the domination count of any element equals to zero, then assign them in another list Q. This list becomes the second nondominated front. Now, for each element of Q, the above method is repeated and the third front is obtained. The above process repeated until all fronts are identified. To maintain the diversity in the population if some solutions may acquire the same rank then the crowding-distance computation requires diversity preservation. It first needs to sort the population according to each objective function value in increasing order of magnitude. Thereafter, for each objective function, the solutions which have the smallest and largest function values are assigned an infinite distance value and rest intermediate solutions are assigned a distance value equal to the absolute normalized difference in the function values of two adjacent solutions. This computation has been continued with other objective functions. The overall crowding-distance value is calculated as the sum of individual distance values corresponding to each objective. Each objective function is normalized before calculating the crowding distance. The algorithm for computing crowding distance is shown in figure 3.7

Algorithm 2 CrowdingDistance(I)

Input: I : Solutions of the front;
Output: I: Assigned crowding distance solutions of the front;

```
l = |I|
for each solution i do
    I[i]_p = 0
end for
for each objective m do
    I = sort(I, m)
    I[1]_distance = I[l]_distance = ∞
    for i=1 to l-1 do
        I[1]_distance = I[1]_distance + (I[i+1].m − I[i−1].m)/(f_m^max − f_m^min)
    end for
end for
```

The flowchart of NSGA-II is illustrated in figure 3.8. At first, a set of parent population is created randomly. This population is sorted using a non-dominated sorting algorithm mention in algorithm 1. Based on their non-domination level the solutions have been assigned a fitness or rank. Then the binary tournament selection is performed to create a mating pool. The winner parent is selected based on rank. If both individuals have the same rank then the individuals having minimum crowding distance is selected otherwise resolved randomly. The individuals in the mating pool have to pass through the process of crossover and mutation

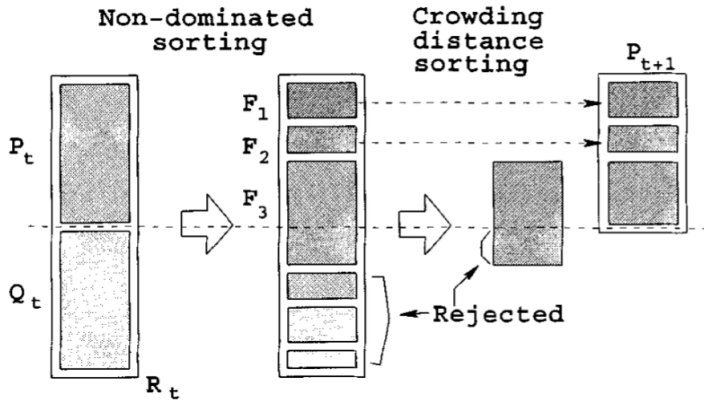

FIGURE 3.8: Flowchart of NSGA-II

to create offspring population of the same size as that of parent population size. The non-dominated sorting is performed on the combined population of parents and offspring. The elitist selection strategy shown in figure 3.8 is used to create a new parent population (same size as the previous parent population) for the next generation. This process is repeated until it satisfies the termination condition (i.e Maximum generation reached).

Chapter 4

Survey of literature

In this chapter, we discuss relevant studies related to different existing methods or algorithms in context of community detection. We have partitioned the relevant existing method in traditional approach and Optimization based approach.

4.0.1 Traditional Approach

In early 1970s, many researchers were attracted toward the community detection problem. They have researched on this problem using graphs [1], [2]. Most of the algorithms were proposed which were based on clustering. These algorithms are called traditional algorithm. We have given some popular traditional algorithms which provide some fundamental concepts of community detection.

Partitional Clustering

Partitional Clustering divides the nodes of the network into g (assumed) clusters by optimizing a loss function. The loss function is based on the distance measure. Some partitional clustering techniques are given which uses a loss functions from different perspective.

- Minimum K-clustering: In this case, the cost function is the maximum distance between two points of a cluster commonly called cluster's diameter .
- K-clustering sum: This concept is similar to minimum K-clustering but the difference is the cost function used. In K-clustering sum the cost function is the average distance between pairs of cluster points instead of the cluster's diameter.
- K-center: In this case, at first a centroid is computed for every cluster. The maximum distance of each node from centroid is computed as the cost function and the clusters and centroids are chosen in such a way that the largest value of diameter should be minimum.
- K-median: it is similar to K-center but, the cost function consists of the average distance instead of maximum distance.

The most popular method in partitional clustering is the k-means clustering [3] which minimizes the intra-cluster distance as a loss function. Another popular method is fuzzy k-means clustering [4], [5]. The fuzzy k-means clustering approach can handle the belonging of a node to more than one clusters.

Hierarchical Clustering

The network or graph may have a hierarchical structure consist of several levels of node clusters. Partitional clustering may not suitable if the size and number of clusters about the given graph is not known. In this situation hierarchical clustering has a very important role to

solve the problem. It is applicable for the graphs which have hierarchical structures [6]. The hierarchical clustering create a binary tree and merges similar clusters based on the similarity between vertices. It does require to provide the number of clusters apriori as compared to partitional clustering. Following are some popular approaches of hierarchical clustering.

1. Agglomerative Algorithms: Agglomerative algorithm is based on bottom up approach.In this algorithm, clusters are combined iteratively if they have a high similarity value (or similarity score) [7]. Agglomerative algorithm initially assumes that each cluster consists of a single object. It then merges the adjacent clusters based on similarity score. are the Some examples of agglomerative hierarchical clustering algorithms are maximal clique and hierarchical link based clustering. The main advantage of these method is that it can generate a small clusters which maybe beneficial in community detection. But this method has a lack of provision for object's relocation if merged incorrectly at an early stage and some times nodes with one neighbor are often classified as an independent cluster which does not make sense in certain cases. It can not be scalable when the points are embedded in space and distance is used as a dissimilarity measure in clustering process. If the distance is not known then the complexity becomes $O(n^2)$ for single linkage and $O(n^2 logn)$ for complete average linkage.

2. Divisive Algorithms: This method is based on top down approach [8]. It initially starts with taking whole samples in a single cluster and then iteratively divide the larger cluster by removing the edges which has low vertex similarity and highest edge betweenness [9]. After a dendrogram is made, communities can be obtained by cutting the tree. The position for cutting is very crucial because if an appropriate cut position cannot obtained then it will result in low quality product. Girvan and Newman proposed a lot of methods based on divisive algorithms [10], [11] for community detection.

Spectral clustering

Spectral clustering method divides the graph into clusters using the eigen vectors of the input data matrix [12]. It transforms the data object into a set of point which are eigen vectors. This conversion presents the internal properties of data sample. Spectral clustering used to cluster data which did not handled by k-means. Donath and Hoffman [13] were the first who contribute in spectral clustering. They used adjacency matrix and similarity matrix of the graph for computing eigen vectors and eigenvalues to partition the graph. In the very same year, Fielder [14] bipartite the graph using the second smallest eigen value of Laplacian matrix. The Laplacian matrix is constructed by the difference of degree matrix and adjacency matrix of a graph. [15], [16] uses spectral clustering method for solving community detection problem and Meila et. al [17] and Ding [18] have used spectral clustering for machine learning problems.

Graph Partitioning

This algorithm divides the vertices into g communities whose size is predefined in a manner to get the minimum number of edges between identified communities [19]. The Kernighan-Lin algorithm [20] is one of the earliest work in this field which uses a heuristic approach for graphs partitioning and still applied in combination with other techniques. It minimizes an objective function which is the difference of the intra community and inter community links. This approach was innovated from the problem of segmenting electronic circuits on boards, in which the vertices in different boards were supposed to be linked with a minimum number of interconnections. In fact,this is an optimization problem. The optimization function may be the difference between the number of edges inside the module and the number of edges

lying between them. Another very known method is the spectral bisection method, which is based on properties of Laplacian matrix spectrum [21]. Apart from this, level structure partitioning, multilevel algorithms, and geometric algorithms are graph partitioning method, whose details can be seen in [20].

4.0.2 Modularity based Optimization Approach

Extremal Optimization(EO)

Extremal Optimizationy is proposed by Boettcher and Percus [22], [95] which is a heuristic based search approach for approximating solutions to hard optimization problems. It is based on non-equilibrium dynamics in which the systems manifesting self-organized criticality (SOC) [96]. A self-organized criticality is a state in which a number of proper solutions come forth actively without tuning the parameters. It optimizes modularity function Q by usind GA as framework having proportional to Q. The fitness of a vertex is calculated as the ratio between vertex modularity and its degree. It is more computationally efficient than simulated annealing but equivalent in performance. Li [97] proposed a method named as pairwise constrained structure-enhanced extremal optimization-based semi-supervised algorithm (PCSEO-SS algorithm) that can solve the problem of false connections along with detecting communities precisely. It can work efficiently if prior information is limited.

Spectral Optimization

This method uses spectral information in the form of eigenvalues and eigenvectors of the modularity matrix that can be used to optimize modularity [23]. To optimize modularity on bisections, Laplacian matrix is replaced with modularity matrix and can be optimized through spectral bisection [11]. If vertices are shifted from one to another community to obtain the increase in highest modularity value or lowest decrease in modularity value which may improve the result. This method can also be useful in greedy algorithms and extremal optimization.

Greedy Optimization

Newman proposed a greedy method based on agglomerative hierarchical approach to maximize the modularity [24]. Clauset et al. [25] showed that the matrix e_{ij} (matrix e_{ij} shows the fraction of edges between clusters i and j of the current partition) used by Newman to compute modularity Q, has a lot of unnecessary operations because of sparse adjacency matrix. They used max-heaps for computing e_{ij} to improve efficiency in algorithmic. This method is much faster than the Newman's greedy approach. The computational cost of the algorithm was O(nlog N). Similarly, Denon et al. [26] proposed an approach for optimizing modularity, which normalizes the modularity variation $\triangle Q$ by the fraction of links incident to one of the two groups to favor small communities. This approach works better than Newman's modularity optimization approach, when community sizes are largely different. Blondel et al. [27] proposed Louvain algorithm which is also a modularity optimization approach based on a heuristic method for detecting communities in the networks of unprecedented size. It is computationally efficient than Newman and Clauset's algorithm. Its computational efficiency depends upon the number of edges in graph and linear in time, i.e., O(m).

Simulated Annealing

Simulated Annealing is probabilistic based optimization method which is usually used for finding near global optima.It is commonly used for maximizing the modularity to get communities in a complex networks. Guimera and Amaral [29] have proposed an algorithm

based on optimization which also used simulated annealing (SA) for the regulation of local search process. This method has good performance in obtaining global solution and it does not require the number of communities. Liu et al. [28] have used k-means algorithms with the simulated annealing. This method detects communities along with their central nodes but this method require prior knowledge of about the number of communities in complex networks.

Genetic Algorithms

Genetic algorithms (GAs) are the search and optimization techniques which mimics the process of biological evolution. GAs can also be applied to optimize the modularity (Q) of the network to identify the community structures which reside in the network. Earlier, GAs were used in graph partitioning [30]. Tasgin et al. [31] was the first who used GA to detect community structure in the complex networks based on optimization of modularity. These algorithms do not need to specify the number of communities apriori. Pizzuti [32] proposed a GA based method named GA-NET which adopted the approach of community score to show the partitioning quality of the social networks. This new concept (Community Score) maximizes the internal links in a community structure. Gong et al. [33] presented a memetic algorithm based on GA named Meme-Net by optimizing the modularity density. In this algorithm, local search climbing strategy was combined with GA search strategy to improve the performance of traditional GAs. Gong et al. [34] were the first to use an evolutionary algorithm for optimizing two contradictory objectives named as negative ratio association and ratio cut. Similarly, Liu et al. [35] and Zeng et al. [36] proposed MOEAs for detecting communities in signed social networks.

4.0.3 Dynamic Algorithm

Random Walk

A random walk is performed on a graph by passing over the nodes randomly. This method is employed to identify the clusters by merging different groups using a bottom-up approach. Zhou [37] represent the distance measured between a pair of edges through the random walk. The distance between any two vertices is measured as the average number of edges made through random walk to reach from one node to the another. Self loops are likely to belong to the same community. Zhou and Lipowsky [38] used biased random walk. In this random walk, the walk is performed on the nodes which have maximum neighbors with the starting node in graphs. The authors proposed a procedure called Netwalk which uses the Brownian movement, that detects communities in this biased random walk. Tis approach is based on agglomerative hierarchical clustering method. In 2005, another algorithm named Walktrap proposed by Pons et al. [39] in which they used modularity value to cut dendrogram using random walk. This random walk is based on similarity between nodes and between the clusters. But this method is computationally expensive having time complexity $O(n^2 logN)$.

Spin models

This models have been used usually in statistical mechanics. Potts model [40] is one of the popular approach in this domain. Reichardt and Bornholdt [93] proposed a community detection method inspired from the idea of super paramagnetic data clustering [41]. They mapped network on the Potts model having a zero temperature q along with the interactions of nearest neighbors. Potts spin variables are assigned to the nodes with community pattern. Later on, Reichardt [42] proposed a spin glass techniques with an assumption that each vertex

should be in a spin state. Apart from its non-deterministic nature, this algorithm has tunable parameters regarding the size of the community.

Synchronization

Basically, the need of synchronization occurs when some part of sysyems or itself systems are interacting units. Synchronization can be applied to clustering problems [43]. If the oscillators are put at the vertices with random phase, they first synchronize the community in which they are placed rather than other communities. If evolution time is permitted then the communities may be recognized by full synchronization in the graph [44]. Arenas et al. [45], who was first to use synchronization in community detection. They showed that the structural scales exposed by synchronization technique which represent the clusters of eigenvalues of the Laplacian matrix of the graph that aids in graph clustering. Boccaletti et al. [92] based on synchronization which uses Kuramoto's model that devised opinion changing rate model [46]. The time complexity of this method is $O(mn)$ and the method shows better performance than the Girvan Newman benchmark. One of the major limitation of synchronization based algorithms is the unreliability in the case of varying size communities.

4.1 Algorithms for Overlapping Communities

Gregory [47] proposed Cluster-Overlap Newman Girvan algorithm (CONGA), a variant of Girvan and Newman traditional method, for overlapping community detection. This method extended with a vertex splitting procedure. In his improved version named as CONGO [48], he used local betweenness to optimize the speed.

There are some popular methods for overlapping community detection.

4.1.1 Local Expansion and Optimization

Baumes et al. [49] proposed a method based on the iterative scan (IS) and rank removal (RaRe) for detecting overlapping communities. RaRe ranks the nodes based on certain condition. A highly ranked nodes are removed continuously to form small disjoint core nodes communities called seed communities for the IS process. IS technique processes a greedy optimization with expanding these seed communities and stop when the density function cannot be improved. Kelley [50] improved IS process and named it CIS by iteratively checking the connectedness. In [89], Chen used the local maximal neighbor's degree of the starting node instead of local modularity to discover community structure.

4.1.2 Clique Percolation Method (CPM)

CPM is based on notion of subgraphs where each node is linked to other in a clique. Palla et al. [51] proposed that the subgraph having high internal edges-density intended to form cliques while inter-community edges do not form cliques. The term K-clique represents the a complete graph having K vertices. It was already assumed that the network consists of K adjacent cliques sharing K-1 nodes in common with each other. Each clique represents a community which shares common node with other communities. A variant of CPM is developed in [52] called sequential clique percolation algorithm (SCP) which is computationally efficient than CPM. It starts from an empty graph and detects K-clique communities by sequentially inserting the edges of the graph taken under consideration. Chakraborty et al. [53] proposed a CPM based method named OverCite which can detect overlapping communities in citations network that contains information about authors, papers, and venues.

4.1.3 Line Graph and Link Partitioning

In this method, graph partitioning is done using link instead of node. In such concept, a node is said to be overlapping if the links connected to it belong to more than one community. Ahn et al. [54] partitioned the links of the network using the hierarchical clustering of edge similarity. Kim et al. [55] extended infomap [57] to partition the links of graph using minimum description length principle and encoded minimum path of the random walk on link network. Evans [56] uses clique concept in line graph for detecting overlapping communities where cliques act as nodes of the weighted graph.

4.1.4 Agent based and Dynamical Algorithm

The Label Propagation Algorithm (LPA) [58] is simple and computationally efficient. In LPA, a community can be formed if nodes having the same label. In each iteration the node in a network is visited and assigned a label based its neighbors' voting and this process repeated till convergence. Gregory [59] extented the LPA named community overlap propagation algorithm (COPRA). This algorithm has ability to detect overlapping communities. Xie et al. [60] also extended LPA and named speaker-listener label propagation algorithm (SLPA) to identify overlapping communities and overlapping nodes. SLPA can be employed to weighted and directed networks by including the interaction rules known as SLPAw.

4.1.5 Fuzzy Detection

Fuzzy Detection methods are label propagation methods which can be extended for overlapping communities in the network by computing fuzzy belonging factor for each node [47] but need prior information of dimension of fuzzy belonging factor. Nepusz et al. [61] developed a fuzzy community detection method with a simulated annealing framework to solve the overlapping community problem as a nonlinear constrained optimization.

4.1.6 Non Negative Matrix Factorization (NMF) Approaches

It is a machine learning algorithm that break up a given feature matrix so that it can find the features of a given structure [62],[63],[64]. Zhang et al. [65] proposed NMF based approach in which the input feature matrix is replaced with diffusion kernel that is Laplacian of given network. This method provides the information about how much a node belongs to a certain community and can detect overlapping communities. Zarei et al. [66] have developed a NMF based approach which replaces input feature matrix with the correlation matrix of column vectors of Laplacian matrix. Psorakis et al. [67] presented a hybrid approach in 2011, that involved Bayesian NMF model to detect overlapping communities in a network. In 2013, [63] was developed that is an alternative of non-negative matrix factorization (NMF).

4.1.7 Recent Developed algorithms

Li et al. [90] proposed a method based on spectral-clustering named an improved multiobjective quantum behaved particle swarm optimization (IMOQPSO) to tackle the overlapping community detection problem. Wen et al. [91] developed a maximal clique-based evolutionary algorithm named (MCMOEA) for detecting overlapping communities. In this method used a new representation scheme based on the maximal clique. Recently in 2017, Zhang [68] proposed a PSO based method which used a mixed representation scheme having room for both overlapping and non-overlapping nodes. This algorithm first detect overlapping nodes based on network structure and then employ a mixed representation scheme with PSO framework for detecting overlapping communities.

Chapter 5

The Proposed Method

In this chapter, we first discuss community detection problem and then describe our proposed methodolgy. A complex network can be represented as a graph denoted by G= (V,E), where $V = \{V_1,V_2,...,V_n\}$ is the set of nodes and $E \subseteq V \times V$ is the set of edges and the aim of community detection is to divide the whole network G into small groups which are also called communities. Let $C = \{C_1,C_2,...,C_k\}$ be the set of all communities reside in G. C_i is said to be a community if it satisfies the following equation:

$$C_i \subset V \text{ and } C_i \neq \phi, i = 1,2,...,k \tag{5.1}$$

We can commonly see that some of the nodes in network have common participation to some communities. These communities are overlapping communities. A community is overlapping or non-overlapping if it satisfies either of the following condition.

$$C_i \cap C_j = \phi, \forall i \neq j \text{ and } i,j \in 1,2,...,k \tag{5.2}$$

$$C_i \cap C_j \neq \phi, \forall i \neq j \text{ and } i,j \in 1,2,...,k \tag{5.3}$$

If a community satisfies equation 5.3 then it is overlapping community otherwise it is non-overlapping community. Figure 5.1 illustrates a network G which has two hidden overlapping communities.

A community detection problem can be transformed into optimization problem. This can be

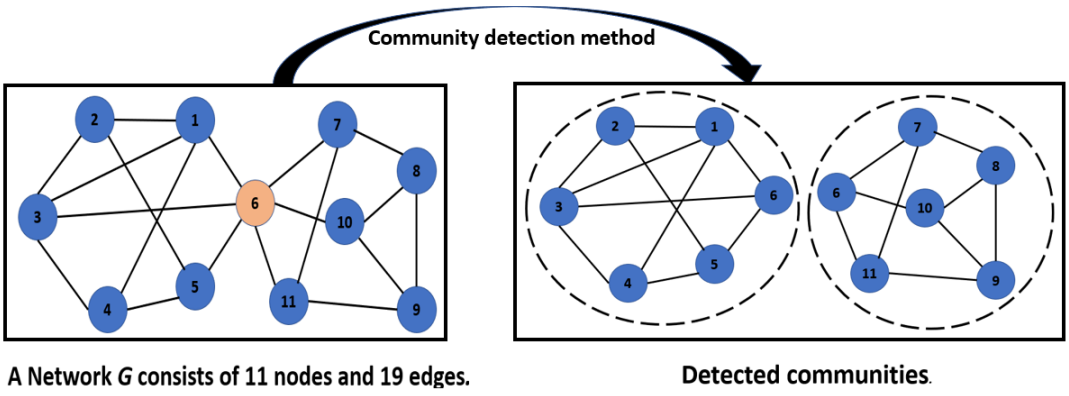

FIGURE 5.1: Detection of communities residing in a network.

done by formulating the function which includes the property of a community. This function will be same as the function involved in optimization problem. Therefore we can say that a community detection problem is transformed to optimization problem and optimizing this function will produce solution

We have already discussed the problem formulation of a community detection problem. Now we discuss our proposed method to solve a community detection problem. At we first need to detect potential overlapping nodes in the network. We have followed the same procedure for overlapping node detection as it is in [68].

5.1 Overlapping Node Detection

In complex networks, we can commonly see that some of the nodes in the network have common participation to some communities. These communities are overlapping communities. Hence, nodes in a network are divided into two parts first is a set of non-overlapping nodes and the second one is a set of overlapping nodes.

- Key Neighbouring Nodes : A key neighbouring node denoted as V_i^{KN} is a neighbour node of V_i, which has the highest number of common neighbouring nodes with V_i.
- Key neighbouring sub-graph:A key neighbouring sub-graph for node V_i is denoted as G_i^{KN} , is the subset of graph G consisting of key neighbouring node (V_i^{KN}) of V_i and common neighbors of V_i and V_i^{KN}.

Algorithm 3 *OverlappingNodeDetection*(**A**, *Num*)

Input: **A** : the adjacent matrix of G; *Num*: the number of nodes in network G;
Output: O; Candidate Overlapping Nodes;

$O \leftarrow \phi$;
$G_{i,1}^{KN}, G_{i,2}^{KN} \leftarrow \phi$; //two key neighboring subgraphs;
for $j = 1$ to 2 **do**
  $V_i^{KN} \leftarrow$ get one key neighboring node of V_i from NB_i;
  $CN_i \leftarrow$ get common neighboring nodes between V_i from NB_i;
  $G_{i,1}^{KN} \leftarrow V_i^{KN} \bigcup CN_i$;
  $NB_i = NB_i - G_{i,j}^{KN}$;
end for
if $LC(G_{i,1}^{KN}, G_{i,2}^{KN}) \leq \eta$ **then**
  $O \leftarrow V_i \cup O$;
end if

In this algorithm for each node V_i, a Key Neighbouring Nodes (V_i^{KN}) and Key neighbouring sub-graph (G_i^{KN}) is generated and if node V_i satisfies the following two conditions then the node V_i is detected as a potential overlapping node.

1. The number of different key neighboring subgraphs of V_i should be greater than 1.
2. There should be spare links between any two key neighboring subgraphs of V_i.

The link closeness(LC) between two neighbouring subgraphs is given in eq. 5.4.

$$LC(G_1^{KN}, G_2^{KN}) = max\left\{\frac{L(G_1^{KN}, G_2^{KN})}{L(G_1^{KN}, G_1^{KN})}, \frac{L(G_1^{KN}, G_2^{KN})}{L(G_2^{KN}, G_2^{KN})}\right\} \tag{5.4}$$

where G_1^{KN}, G_2^{KN} are two subgraphs and L is a function defined in eq. 5.5 is the number of links between two subgraphs.

$$L(G_1^{KN}, G_2^{KN}) = \sum_{j \in G_1^{KN},\ k \in G_2^{KN}} A_{ij} \tag{5.5}$$

The links between the key neighboring subgraphs G_1^{KN} and G_2^{KN} are spare if the link closeness is less than a specified threshold η.

5.2 Chromosome Representation

The nodes in a complex network may have some set of overlapping nodes. So we need a valid representation for overlapping nodes to apply. Therefore we have used a mixed representation technique for representing chromosomes which are based on [mixed 36]. The aim of this mixed representation scheme is to provide room for both overlapping and non-overlapping nodes simultaneously. In this mixed representation technique, we have used the size of the chromosome equal to the number of nodes in the network. For the chromosome representation of overlapping nodes, we have used -1 and 0 to show overlapping node status. -1 indicates that the potential overlapping node is suppressed state i.e it will act as a non-overlapping node and 0 indicates that it is in active state i.e it will behave as an overlapping node. The label used for the potential overlapping node having status -1 followed with most occurred its non-overlapping neighbor's label and for status 0 it is labeled with all its distinct non-overlapping neighbor's label whose appearance is greater than one following its status. If its distinct non-overlapping neighbor's label equals to one then randomly label with any one label. For non-overlapping nodes, the label used in chromosome with any one of the number between 1 to $|C|$ where $|C|$ is the number of communities in the network. Figure [ref] shows the overall community detection phase. In figure 1a, network structure with 11 nodes and 16 edges are shown and the corresponding chromosome representation is shown in fig 1c.

5.3 Objective Functions

Our proposed method is based on a multi-objective genetic algorithm(MOGA). In this work, we have used two objective functions. The first objective function maximizes the intra-community link density whereas the second objective function minimizes the inter-community link density. These objective functions are mention in eq. 5.6 and eq. 5.7 respectively.
The interpretation of Eq. 5.6 and its description is given below.

$$obj1 = \sum_{l=1}^{|Com|} \frac{\sum_{m=1}^{|Com_m|} (\mu_{ij} * Int\_Deg_{lm})}{|Com_m|} \tag{5.6}$$

Where, $|Com|$ is the number of community in network, μ_{lm} is the fuzzy membership value of l^{th} node of m^{th} community which is given in eq. 5.8 and Int_Deg_{lm} represents the internal degree of l^{th} node of m^{th} community. $|Com_m|$ denotes the total elements in m^{th} community.

The representation of eq. 5.7 and its detailing is given below:

$$obj2 = \sum_{l=1}^{|Com|} \frac{\sum_{m=1}^{|Com_l|} Deg\_CC_{ml} - Int\_Deg\_CC_{ml}}{|Com_l|} \tag{5.7}$$

Here, $|Com|$ denotes the total number of communities and $Int\_Deg\_CC_{ml}$ represents the sum of internal degree (Int_Deg) and internal clustering coefficient (Int_CC) of l^{th} node of m^{th} community and its equation is shown in equation. Similarly, Deg_CC_{kl} interprets the sum of degree and clustering coefficient of node k in l^{th} community.

The power for maximizing intra-community link density is given in Eq. 5.6 that greatly support the dense connections within the communities. We have observed that nodes in a

network are commonly influenced by its neighbors. We have adopted degree and clustering coefficient as nodal attributes which have key importance in computing fuzzy membership values. It has been used because some nodes in network might have several association with communities. This fuzzy member value consists of internal degree and internal clustering coefficient and the corresponding equation is illustrated in equation 5.8. In this function, the term μ_{lm} represents the fuzzy membership value of l^{th} node of the m^{th} community that is treated as a measure of the weighting factor for the corresponding nodes. If the membership value of a node is higher, its belonging in that communities will be higher. The numerator term *Int_Deg* in the first objective function (eq. 5.6) represents the internal degree and it is used as the measure for the quality of the community. Higher this value, higher will be the quality of the community. The term $|Com_m|$ interprets as the number of nodes in m^{th} community. On dividing the numerator term (i.e. the sum of the product of membership and internal degree) with the denominator (i.e $|Com_m|$) will give rise to the link compactness or link density. An increase in the link compactness, there will also increase in the quality of the communities. This link compactness strongly satisfies the high connections between the members of the community.

The second objective function is already represented in Eq. 5.7 that is used to minimize inter-community link density. The term $(Deg\_CC - Int\_Deg\_CC)$ of equation 5.7 is the numerator difference term that represents the difference between the sum of degree and clustering coefficient and the sum of internal degree and internal clustering coefficient. In other word, it is the sum of degree difference(i.e difference in original degree and internal degree) and clustering coefficient difference(i.e difference in actual clustering coefficient and internal clustering coefficient). The term $|Com_m|$ interprets it as the number of all members in m^{th} community. For the formation of density the numerator of this objective function is divided with $|Com_m|$ will represent the external link density. The decrease in this value will lead to an increase in the quality of the communities.

The representation and calculation of fuzzy membership values is illustrated below.

$$\mu_{km} = \sum_{m=1}^{|Com|} \sum_{k=1}^{|V|} \frac{\left(\frac{Int\_Deg_{km}}{Deg(V_k)} + \frac{Int\_CC_{km}}{CC(V_k)}\right)}{\sum_{n=1}^{|Com|}\left(\frac{Int\_Deg_{kn}}{Deg(V_k)} + \frac{Int\_CC_{kn}}{CC(V_k)}\right)} \tag{5.8}$$

Here, μ_{km} is the fuzzy membership value for node V_k in community Com_m. $|Com|$ and $|V|$ represents is the number of community and the number of nodes in network G respectively. The term Int_Deg_{ij} represents the internal degree while and Int_CC_{ij} represents the internal clustering coefficient of k^{th} node of m^{th} community . $Deg(V_k)$ represents the degree of node V_k while and CC_{V_k} clustering coefficient of node V_k.

5.4 Population Initialization

After the encoding of the chromosome, we need to initialize the population. In the population-based method the initialization of population is commonly done randomly. In our method, for each non-overlapping node, we randomly generate a label between 1 to $|C|$ where $|C|$ is the number of community in the network already known and for overlapping nodes we randomly label with either -1 or 0 randomly. But for population-based methods, random initialization of the population might generate low-quality solutions. So some guidance is needed to generate high quality solutions. For this purpose, after random initialization as done aforesaid, we used neighbor based initialization for each non-overlapping node having a degree greater than an average degree in network replaced with the most appeared label of its neighbors. For overlapping nodes having -1 is labeled with most occurred neighbor's

label with following -1 at the start. For status 0, we keep this status same and label with all distinct neighbor's label whose occurrence is greater than one. If all the occurrence is one then label with anyone. For overlapping node, if all neighbors are overlapping then keeping its status, label randomly with anyone in range 1 to $|C|$. This strategy is performed on nodes in the order they have appeared in the chromosome.

5.5 Genetic Operators

The already existing community detection algorithm based on GA is not efficient enough to give a relevant solution. The main reason for their inefficiency is their genetic operator i.e crossover and mutation operator. The crossover operator provides low convergence and mutation operator provides random changes in the gene. This generates a poor solution. So in this thesis, we have used one point crossover and a new mutation operator. An updation method has also been proposed in this work for updating the chromosome after crossover or mutation.

5.5.1 Selection

In our method we have used binary tournament selection and elitist selection strategy. In Binary tournament selection strategy, two random chromosomes from the population are selected to compete and the chromosome which wins is selected for the formation of mating pool. This selection strategy provides chance for the chromosomes having low fitness to be selected in mating pool. The selection of winner chromosome depends on Pareto front ranking which is based on non-dominated sorting. To resolve the problem of same Pareto front ranking crowding distance is used. If the crowding distance is also same then it solved randomly. After the generation of solutions, the elitist selection strategy comes into attack. This selection strategy is used to select chromosomes from combined child and parent population of size $2N$ to a new population for upcoming generation of size N. The elitist strategy is a selection strategy which keep track for the best solution or some sets of best solutions. Here we have to select N best solutions out of $2N$. The way is to first select first front then second front, third front and so on. If the currently selected front cannot be placed for the population of next-generation then the chromosomes from this front are selected based on crowding distance to make the size of the population N.

5.5.2 Crossover

In our proposed method we have used one point crossover which is simple and easy for implementation. One point crossover produces offspring similar to parent. The implementation of crossover is mentioned in algorithm 4. In this method if the randomly generated number between 0 and 1 is less than or equal to crossover probability(CX_P) then this process further executed otherwise return the same. The position for crossover is randomly selected any position except starting position and ending position. The process of crossover operation is shown in figure 5.2. After getting offspring a indicator is used namely "indicate". This variable decides the updation is for crossover or mutation. We have assigned it to a value -1 that will indicate the updation needed for crossover. The children are updated using update method whose procedure is given in algorithm 6. At the end if the number of communities obtained in parent and child are equal then the child is added in $new\_offspring$, else parent is added to $new\_offspring$.

Algorithm 4 *Crossover*(a, b, n, G)

Input: a, b : Individualss; n: Total number of communities; G : Nodes of the network G;
Output: $new\_offspring$: Offspring Individuals;

```
parent ← [a,b];
if rand(0,1) ≤ CX_P then
    offspring ← perform one point crossover to get children from parent
    new_offspring ← ϕ
    for i = 1 to |offspring| do
        indicator= -1 //indication for updation after the crossover operation.
        offspring_i ← Update(offspring_i, indicator, n, G)
        n_com ← number of communities in offspring_i
        if n_com == n then
            new_offspring ∪ offspring_i
        end if
    end for
    if |new_offspring| ≠ |parent| and |new_offspring| > 0 then
        offspring ← randomly chose a chromosome from parent
        new_offspring ∪ offspring
    end if
else
    new_offspring = parent
end if
```

5.5.3 Mutation

In most of the Genetic Algorithm based methods, the mutation is random. But to control its randomness over a global view, we have restricted it to the local view. In this method we have used a mutation operator that has two parts, one is random based and the other is neighbor based. In our proposed mutation operator, at first, if the randomly generated number is less than mutation probability(MUX_P) then steps are further executed otherwise leave the same. The mutation position is randomly chosen at any position in the chromosome. If the selected position belongs to the overlapping nodes then it comes in the category of random mutation where its status is replaced with -1 if previously it was 0 and vice versa. If the selected position belongs to the nonoverlapping nodes then it will belong to any of the two categories (random based and neighbor based) having the same preferences. If it belongs to a randomly based category then the label of the chromosome at mutation position will be replaced with any one of the numbers chosen randomly from 1 to n where n is the number of community in the network. In the case of neighbor based, the label of the chromosome is replaced with most appeared label among its non-overlapping neighbor's label. After such a change, the chromosome undergoes through updation operation. If the modified chromosome contains the total number of communities as that of the parent, then the mutant chromosome will result in this mutation operation otherwise parent.

5.6 Updation

After the implementation of the crossover and mutation operator, it is the duty of the update method to modify the chromosome. After crossover and mutation operation, the label of nodes in the chromosome might be changed. So we need a method to modify the chromosome. This method is mainly applied for overlapping nodes as its status may change or there may be a change in neighbors label. The procedure for updating the chromosome

Algorithm 5 *Mutation*(b, n, G)

Input: b : Chromosome; n: Total number of communities; G: Network or Graph ;
Output: a: Mutant Chromosome;

if $rand(0,1) \leq MUX\_P$ **then**
  $pos \leftarrow$ randomly get a position between start and end of chromosome a;
  if $pos \in overlapping\ node's$ position **then**
    exchange 0 $\leftrightarrow$-1 in label of a at the beginning of pos;
  else if $rand(0,1) \leq 0.5$ **then**
    label the chromosome b at pos with randomly select number between 1 and n;
  else
    label the chromosome b at pos with most repeated label of neighbors;
  end if
  $b \leftarrow Update(b, pos, n, G)$;
  $n_com \leftarrow$The number of community in updated a;
  if $n! = n\_com$ **then**
    leave chromosome same as it was before mutation;
  end if
else
  leave the chromosome same as before
end if

is given in algorithm 6. From figure 5.4 we can observe that the overlapping nodes can easily be verified in chromosome by checking the size of the gene because the label for overlapping node in chromosome has the status "0" or "-1" which is an additional label than community belonging label. Thus the size of the gene in chromosome id greater than one is the corresponding label for overlapping node. There is a variable namely "indicate" which indicates that the updation needed for crossover or mutation. If its value is "-1" then it is for crossover and if its value is any positive number ranging from 1 to V then it is for mutation. The corresponding positive number is the position in the chromosome at which mutation is applied. If the position belongs to the overlapping node's position then the updation occurs at the corresponding gene in chromosome otherwise not. The process of applying updation on overlapping nodes label in the chromosome, the following are the cases that we consider.

1. If the nodes and its all neighbors are overlapping then randomly label the chromosome at corresponding node's position with anyone ranging from 1 to n (total number of communities in the network) and keeping the start status at beginning.

2. If its all neighbors are not overlapping and its status is "-1" then replace the label with the highest occurrence of its non-overlapping neighbor's label in chromosome and retain the start status at the beginning.

3. If its all neighbors are not overlapping and its status is "0" then replace the label with all its distinct non-overlapping neighbor's label in chromosome whose appearance is greater than one. If the appearance is one for all distinct non-overlapping neighbors then label the chromosome for the corresponding node with anyone. Retain the start status at the beginning.

From Figure 5.4, we can see that the node 6 is the overlapping node because its size is greater than one and the label for node 6 is not accurately labeled in figure a. But during updation the label of node 6 has the status "-1" and the occurrence of label "1" is 4 and that of "2" is 6. So the correct label should be "2" because its occurrence is 6 which is greater than the

Algorithm 6 *Update* (a, pos, n, G)

Input: b : Chromosome ; pos: position related to crossover or mutation updation; n: Number of communities; G:Network or graph ;
Output: a: updated chromosome;

$V \leftarrow$ get nodes of G
$N\_U.O \leftarrow \phi, \phi$
for $i = 1$ to $|b|$ **do**
  if $|b_i| \geq 2$ **then**
    $O \leftarrow O \cup V_i$
  end if
end for
$G.adj \leftarrow$ Adjacency matrix of graph G
if $pos > 0$ and $V_{pos} \in O$ **then**
  $N\_U \leftarrow V_{pos} \cap O$
else if $pos < 0$ **then**
  $N\_U \leftarrow O$
end if
for each node$nd \in N_U$ **do**
  $index \leftarrow$ get node index for node nd
  if all neighbors of $nd_i \in O$ **then**
    $a_{index} \leftarrow$ change the label with randomly chosen number ranging [1,n] and reserve the start status as it was before modification.
  else if b_{index} contain 0 **then**
    $b_{index} \leftarrow$ reserve the first element of b_{index}, and change the remaining part with all the distinct its non-overlapping neighbors label that appeared more than once, otherwise exists randomly label with one of its non-overlapping neighbor.
  else if b_{index} contain -1 **then**
    $b_{index} \leftarrow$ reserve the first element of b_{index}, and change the remaining part with the most occurred label of its non-overlapping neighbor.
  end if
end for

number of times the occurrence of "-1". So after updating the chromosome the label of node 6 replaces "1" with "2" and keeping start status "-1" same.

5.7 Procedure of our proposed method

In this section, we have combined the sequence of techniques used in our proposed methodology in the form of flowchart which is shown in Figure 5.5. The proposed methodology is based on the Multi-Objective Genetic Algorithm. We have adopted two objective functions, one is intra-community density maximization and other is inter-community density minimization. The description of these objective functions is mentioned in Eq. 5.6 and eq. 5.7 respectively. To find the Pareto optimal solution set, NSGA-II has been used as a supporting structure. At first, overlapping node detection is done, then the chromosome is encoded using mixed representation in order to generate an initial population such that the chromosome does not produce duplicate communities. This population initialization step is created based on neighbors. After the initialization of the population, fuzzy membership values are assigned for each node in the network. the fuzzy membership value is estimated using internal degree and internal clustering coefficient. After this step, each chromosome is assigned fitness values to each objective function. Then the solutions are ranked using non-dominated

sorting and the crowding distance. The non-dominated sorting is done on conflicting objective functions (eq. 5.6 and eq. 5.7). Then the solutions pass through different stages of genetic operations like selection, crossover, and mutation to generate children solutions do not have duplicate communities. After the genetic operations on solutions, the solution is updating using the update method. The process of genetic operations and updating are done on each generation. The newly generated child population is merged with the parent population and then the combined population having size double the parent population are ranked using non-dominated sorting and crowding distance. The process stops when the current generation reaches a specified maximum generation, then the best solution from the first Pareto front having maximum gNMI and Modularity value based on equations (6.2) and (6.1) are obtained respectively.

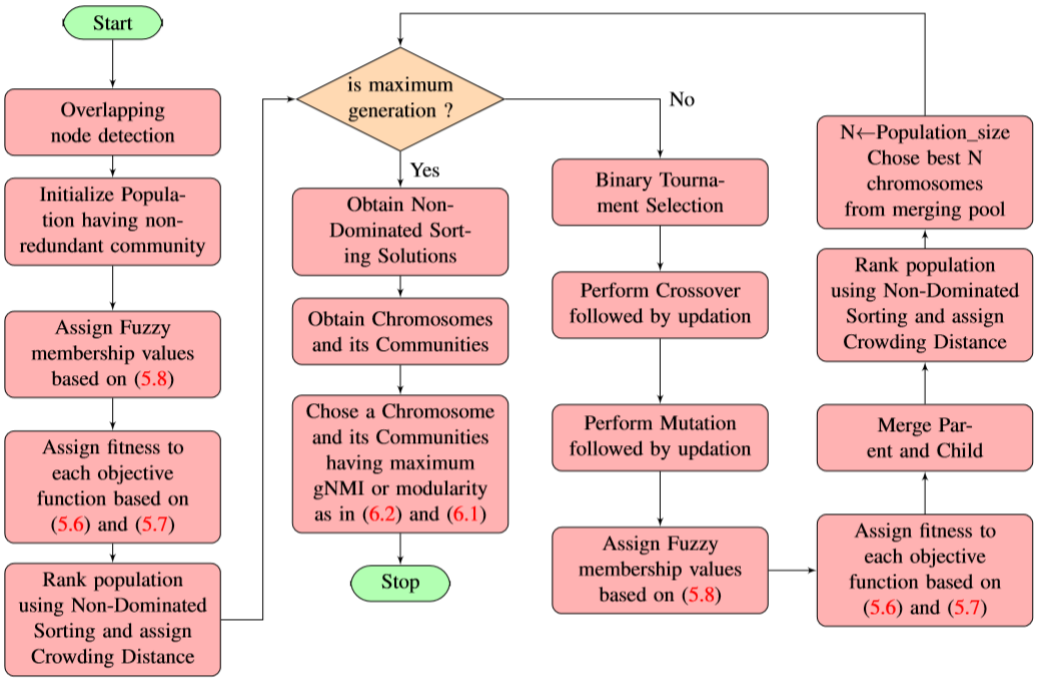

FIGURE 5.5: Flow chart of our proposed method.

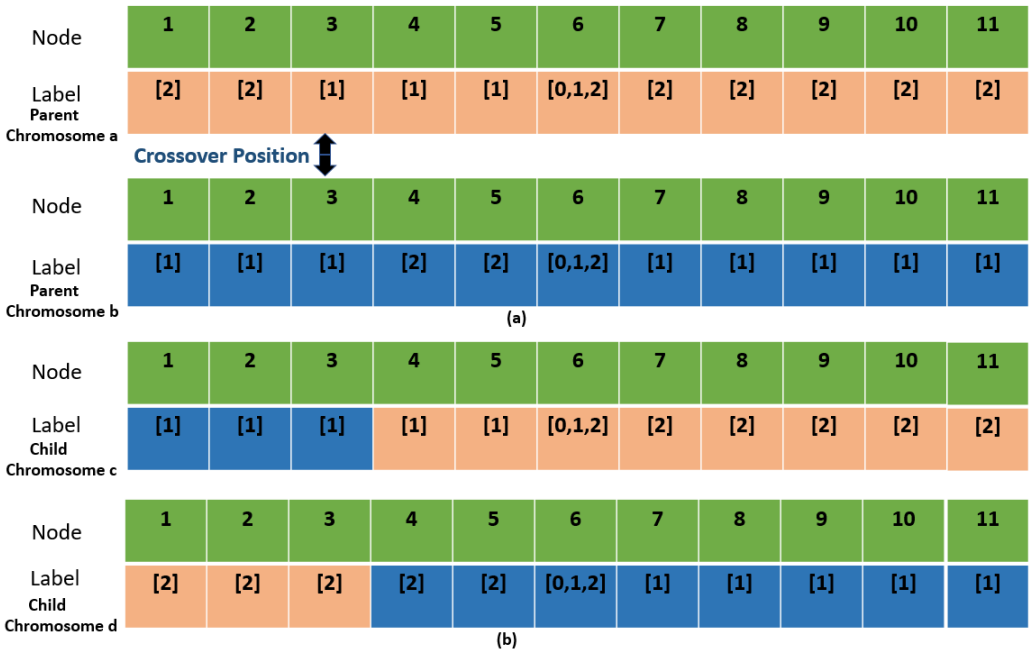

FIGURE 5.2: Crossover between chromosome a and b.

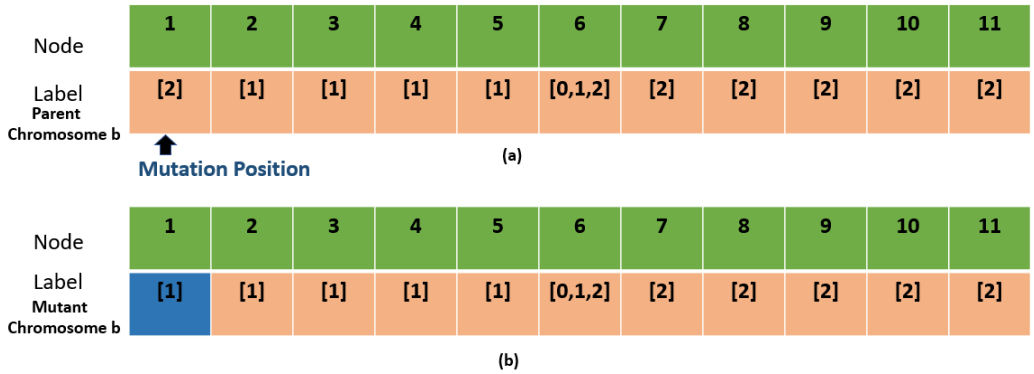

FIGURE 5.3: Mutation of chromosome b.

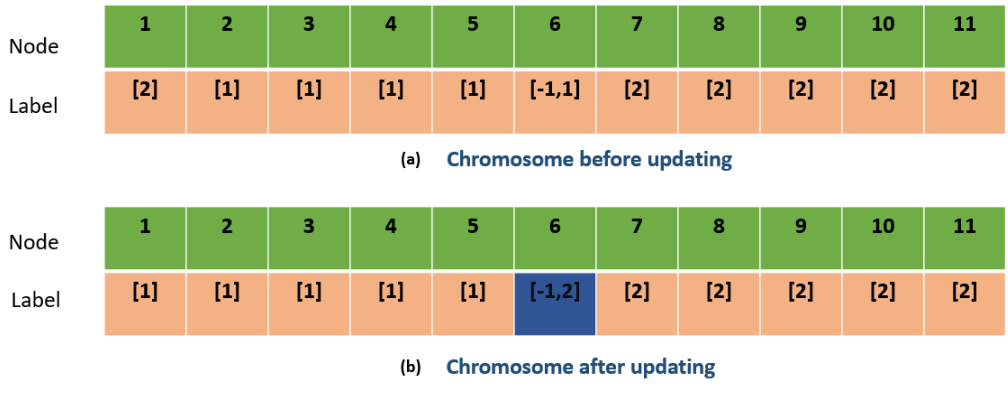

FIGURE 5.4: Updation of chromosome.

Chapter 6

Experimental Results and Discussion

6.0.1 Experimental set up

For a fair comparison, both the population size and the maximum number of generations is set to 100. The threshold η for controlling the number of candidate overlapping nodes has been taken as 0.1. The crossover probability and mutation probability are set to 0.8 and 0.1 respectively. The experimental results for all algorithms are obtained by taking the average over 20 independent runs. All the experiments are carried out on DELL INSPIRON 15 3000 SERIES, CPU @ 2.50GHz 2.70GHz, 4-GB RAM, Windows 10 operating system and Python programming language.

Real World Networks

In our work, we have employed our method on four real-world networks having different community structures. All these networks are well known for their ground truth communities. These networks are easily available on the internet. A brief discussion on the real world networks is given below.

The Zachary's Karate Club is a real-world social network created by Wayne Zachary [73] in the 1970s. This network contains 34 nodes having 74 interconnecting edges over a period of two years. This network is divided into two parts due to an argument between the administrator and instructor of the karate club.

The Bottlenose Dolphins network [74] is also a social network of bottlenose dolphins created by David Lusseau and his co-working members. These Bottlenose Dolphins were seen during 1994-2001 in Doubtful Sound, New Zealand. This network consists of 62 dolphins with 159 frequent associations among them and can naturally separate them into the community of male and female dolphins.

The third network is the American political books network [75] consisting of 105 nodes and 441 edges created by Krebs where node indicates US Political book and the edge represents regularly purchased books by the same customer sold online on Amazon.com. The sold books are separated into "liberal", "natural" and "conservative" disjoint groups.

The last real-world network is American College football games between Division IA colleges. This network[11] is formed by Girvan and Newman during the regular season fall 2000 which consists of 115 teams with 616 regular season matches between the teams. This network is divided into 12 communities.

Evaluation Metrics

In our work, we have taken two standard metrics, gNMI and extended modularity, for measuring the quality of a community. The description of these metrics is given below:

Modularity: The extended modularity is proposed by Nicosia [76] which is suitable for overlapping as well as non-overlapping communities. The equation of this metric is given in

TABLE 6.1: gNMI value-based comparison of eight algorithms on four real-world networks

Network	*Metric*	*our*	*MR-MOEA*	*IMOQPSO*	*MEAs SCN*	*MCMOEA*	*Zhang*	*LMD*	*NMF*
Karate	gNMI max	**1**	**1**	0.708	0.383	0.918	0.513	0.513	0.837
	gNMI avg	**1**	**1**	0.698	0.375	0.890	0.496	0.447	0.837
	std	0	0	0.024	0.042	0.069	0.052	0.104	0.0
Dolphin	gNMI max	**1**	**1**	**1**	0.421	0.473	0.293	0.611	0.907
	gNMI avg	**1**	**1**	0.756	0.412	0.342	0.277	0.456	0.907
	std	0	0	0.475	0.017	0.161	0.089	0.132	0
Football	gNMI max	0.922	0.803	0.809	**0.927**	0.712	0.761	0.783	0.793
	gNMI avg	**0.897**	0.803	0.798	0.788	0.696	0.757	0.762	0.793
	std	0.0279	0	0.015	0.320	0.037	0.007	0.028	0
Polbook	gNMI max	**0.564**	0.149	0.432	0.482	0.104	0.137	0.137	0.388
	gNMI avg	**0.479**	0.139	0.389	0.416	0.098	0.093	0.118	0.388
	std	0.066	0.014	0.032	0.062	0.008	0.059	0.017	0

Eq. 6.1.

$$Q_{ov} = \frac{1}{2m} \sum_{j=1}^{|Com|} \sum_{j \in Com_i, k \in Com_i} \frac{1}{O_j O_k} \left[A_{ij} - \frac{deg(i)deg(k)}{2m} \right] \tag{6.1}$$

Here, Q_{ov} represents modularity value lies between -1 to $1]$ (including -1 and 1),m is the total number of edges in the network, A is the adjacency matrix, $|Com|$ gives the total occurrence of communities, Com_i represents i_{th} community, O_j gives the occurrence of node j distinct communities, and d_j represents the degree of node j.

Generalized Normalised Mutual Information (gNMI) [77]: It is one of the widely used metrics which is used to validate the quality of solution for detected communities against the ground truth. gNMI is defined below in equation 6.2.

$$gNMI(A,B) = \frac{-2\sum_{i=1}^{C_A}\sum_{j=1}^{C_B} C_{ij}\, log\frac{C_{ij}\, N}{C_i\, C_j}}{\sum_{i=1}^{C_A} C_i\, log\frac{C_i}{N} + \sum_{j=1}^{C_B} C_j\, log\frac{C_j}{N}} \tag{6.2}$$

Here, $gNMI(A,B)$ represents NMI value between division A and B, C_A(C_B) represents the number of communities in division A (division B), C represents the confusion matrix, C_{ij} represents the count of common nodes between communities i and j in division A and B respectively, C_i is the sum of elements of C in row i and C_j is the sum of elements of C in column j. N denotes the number of nodes.

6.1 Experimental Results

In our work, we have employed our proposed method on four real-world networks namely Karate, Dolphin, Polbook, and Football. We have compared our method with seven popular algorithms based on extended modularity and gNMI.

TABLE 6.2: Extended modularity value-based comparison of eight algorithms on four real-world networks

Network	*Metric*	*our*	*MR-MOEA*	*IMOQPSO*	*MEAs SCN*	*MCMOEA*	*Zhang*	*LMD*	*NMF*
Karate	Q_{ov} max	0.210	**0.229**	0.213	0.204	0.210	0.216	0.216	0.205
	Q_{ov} avg	0.210	**0.223**	0.208	0.18	0.208	0.212	0.204	0.205
	std	0	0.007	0.004	0.022	0.002	0.005	0.024	0.0
Dolphin	Q_{ov} max	0.200	**0.271**	0.264	0.221	0.206	0.261	0.261	0.200
	Q_{ov} avg	0.200	**0.264**	0.258	0.201	0.198	0.251	0.194	0.200
	std	0	0.011	0.008	0.017	0.049	0.089	0.102	0
Football	Q_{ov} max	0.300	**0.306**	0.243	0.226	0.279	0.282	0.284	0.303
	Q_{ov} avg	0.292	**0.303**	0.235	0.207	0.274	0.271	0.246	**0.303**
	std	0.007	0.005	0.014	0.020	0.087	0.011	0.90	0
Polbook	Q_{ov} max	**0.267**	**0.267**	0.244	0.246	0.228	0.237	0.263	0.259
	Q_{ov} avg	0.242	**0.265**	0.241	0.216	0.222	0.225	0.241	0.259
	std	0.036	0.005	0.004	0.042	0.011	0.012	0.071	0

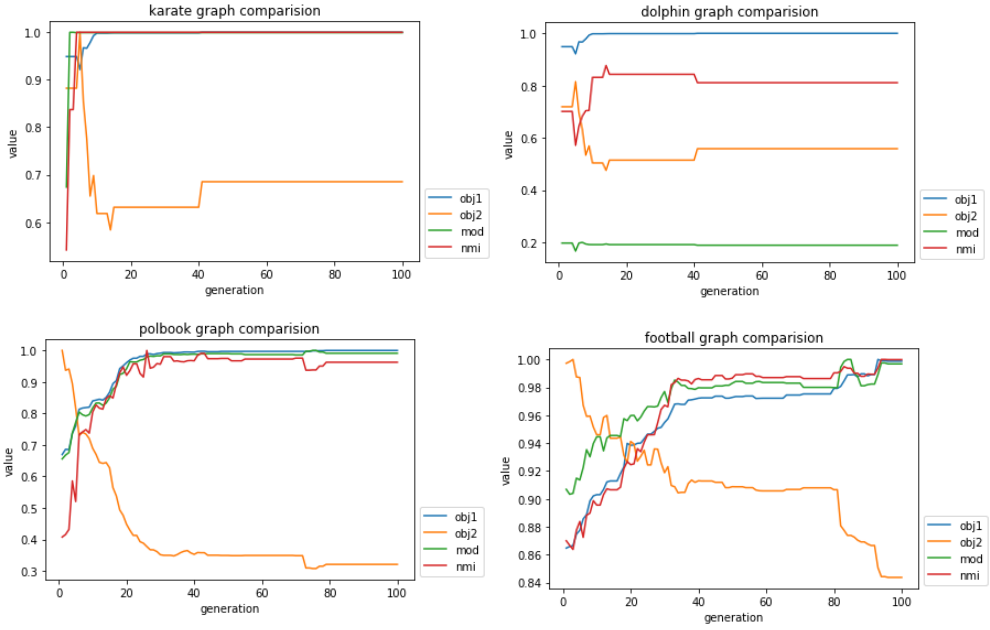

FIGURE 6.1: Comparison of random population initialization and random mutation based algorithms with proposed method based on nmi and modularity

In this work, our proposed method has generated better results than other methods on karate, dolphin, football, and polbook networks in terms of gNMI (*NMI_max* and (*NMI_avg*) over 20 independent iterations which are shown in Table 6.1. The maximum gNMI value has been obtained by our method is 0.922 on football network which is a little bit less than *MEAs_SCN* (0.927), however, it has been ranked second-best among all. Moreover, on the football network, our method has obtained the best average gNMI value among all other methods. Table 6.1 is evidence that our proposed method outperforms the others in terms of gNMI value. While the experimental results on Table 6.2 overviews that our proposed method has shown the comparative result in terms of modularity.

Discussion

In our work, we have developed a community detection method based on a multi-objective evolutionary algorithm and for supporting framework, we have used NSGA-II. NMI and

modularity are used as metrics for validating the detected communities by the algorithms. We have developed two objective functions 5.7 and 5.7. The convergence of both objective functions along with the modularity and NMI is shown in Figure 6.1. These are the normalized average value which is customized by dividing the maximum value obtained on averaging the value in Pareto front over the generation. There are basically two strategies for detecting communities, network structure, and nodal properties. We have hybridized both strategies, overlapping node detection [68] depends on the network structure while computation of fuzzy membership value relays on nodal properties (degree and clustering coefficient as nodal properties). We have also used neighbor based strategy in population initialization, mutation operator and updation method. These all methods together able to grasp the network structure and produce a solution(s) near actual ground truth. Table 6.1 is the evidence for producing our resultant communities are actual or near to actual due to the adoption of the neighbor based strategy.

Chapter 7

Conclusion and Scope for Future Works

In this chapter, we first discuss the conclusion of our proposed method and in the next section, we will light on its scope for future exploration.

7.1 Conclusion

In our work, we have proposed a method based on a multi-objective genetic algorithm for detecting overlapping communities. We have taken two objective functions, one for maximizing internal edge density and other for minimizing external edge density, which strongly satisfies the properties of a community. Apart from this, a fuzzy membership value has been given to each node in the network. This utilizes the benefits of nodal properties of a node in the network. We have adopted a neighbor based approach in population initialization, mutation, and updation. One point crossover operator which includes updation policy used for high convergence and mutation operator has been proposed for some directional change (i.e sometimes random based and sometime neighbor based). Updation method is used for modifying the chromosome both in crossover and mutation. These methods together properly handle the network structure. Experimental results on four real-world networks shown in Table 6.1 indicates that our proposed method shows the higher gNMI value which indicate that the proposed method produces results nearer to actual communities and better in term of gNMI and comparative in term of modularity. However, our proposed method suffers from limitations like require the number of communities in advance and the length of the chromosome equal to the number of nodes in the network. Overall, our method has performed better in terms of gNMI compared to rest methods.

7.2 Scope for Future Works

In the future, we may rectify the limitation of our method and make it capable of generating communities without prior knowledge. It can also be extended to a large dynamic network. In the future, we may apply more recent evolutionary methods like NSGA-III [71] to solve highly complex networks having billions of nodes.

Bibliography

[1] S. Fortunato, “Community detection in graphs,” Physics reports, vol. 486, no. 3-5, pp. 75–174, 2010.

[2] C. Wang, W. Tang, B. Sun, J. Fang, and Y. Wang, “Review on community detection algorithms in social networks,” in 2015 IEEE International Conference on Progress in Informatics and Computing (PIC). IEEE, 2015, pp. 551–555.

[3] J. MacQueen et al., “Some methods for classification and analysis of multivariate observations,” in Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, vol. 1, no. 14. Oakland, CA, USA, 1967, pp. 281–297.

[4] J. C. Dunn, “A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters,” 1973.

[5] J. C. Bezdek, Pattern recognition with fuzzy objective function algorithms. Springer Science & Business Media, 2013.

[6] L. Zhang, Q. Ye, Y. Shao, C. Li, and H. Gao, "An efficient hierarchy algorithm for community detection in complex networks," Mathematical Problems in Engineering, vol. 2014, 2014.

[7] O. Maqbool and H. A. Babri, “The weighted combined algorithm: A linkage algorithm for software clustering,” in Eighth European Conference on Software Maintenance and Reengineering, 2004. CSMR 2004. Proceedings. IEEE, 2004, pp. 15–24.

[8] M. Roux, “A comparative study of divisive hierarchical clustering algorithms,” arXiv preprint arXiv:1506.08977, 2015.

[9] A. Morvan, K. Choromanski, C. Gouy-Pailler, and J. Atif, “Graph sketching-based massive data clustering,” arXiv preprint arXiv:1703.02375, 2017.

[10] M. Girvan and M. E. Newman, “Community structure in social and biological networks,” Proceedings of the national academy of sciences, vol. 99, no. 12, pp. 7821–7826, 2002.

[11] M. E. Newman, “Modularity and community structure in networks,” Proceedings of the national academy of sciences, vol. 103, no. 23, pp. 8577–8582, 2006.

[12] B. Auffarth, “Spectral graph clustering,” Universitat de Barcelona, course report for Technicas Avanzadas de Aprendizaj, at Universitat Politecnica de Catalunya, 2007.

[13] W. E. Donath and A. J. Hoffman, “Lower bounds for the partitioning of graphs,” in Selected Papers Of Alan J Hoffman: With Commentary. World Scientific, 2003, pp. 437–442.

[14] M. Fiedler, “Algebraic connectivity of graphs,” Czechoslovak mathematical journal, vol. 23, no. 2, pp. 298–305, 1973.

[15] A. Pothen, H. D. Simon, and K.-P. Liou, “Partitioning sparse matrices with eigenvectors of graphs,” SIAM journal on matrix analysis and applications, vol. 11, no. 3, pp. 430–452, 1990.

[16] S. T. Barnard, A. Pothen, and H. Simon, “A spectral algorithm for envelope reduction of sparse matrices,” Numerical linear algebra with applications, vol. 2, no. 4, pp. 317–334, 1995.

[17] M. Meila and J. Shi, “A random walks view of spectral segmentation,” 2001.

[18] C. Ding, “A tutorial on spectral clustering,” in Talk presented at ICML.(Slides available at http://crd. lbl. gov/ cding/Spectral/), 2004.

[19] A. Pothen, “Graph partitioning algorithms with applications to scientific computing,” in Parallel Numerical Algorithms. Springer, 1997, pp. 323–368.

[20] B. W. Kernighan and S. Lin, “An efficient heuristic procedure for partitioning graphs,” Bell system technical journal, vol. 49, no. 2, pp. 291–307, 1970.

[21] D. A. Spielmat and S.-H. Teng, “Spectral partitioning works: Planar graphs and finite element meshes,” in Proceedings of 37th Conference on Foundations of Computer Science. IEEE, 1996, pp. 96–105.

[22] S. Boettcher and A. G. Percus, “Extremal optimization for graph partitioning,” Physical Review E, vol. 64, no. 2, p. 026114, 2001.

[23] M. Chen, K. Kuzmin, and B. K. Szymanski, “Community detection via maximization of modularity and its variants,” IEEE Transactions on Computational Social Systems, vol. 1, no. 1, pp. 46–65, 2014.

[24] M. E. Newman, "Fast algorithm for detecting community structure in networks," Physical review E, vol. 69, no. 6, p. 066133, 2004.

[25] A. Clauset, M. E. Newman, and C. Moore, "Finding community structure in very large networks," Physical review E, vol. 70, no. 6, p. 066111, 2004.

[26] L. Danon, A. Díaz-Guilera, and A. Arenas, "The effect of size heterogeneity on community identification in complex networks," Journal of Statistical Mechanics: Theory and Experiment, vol. 2006, no. 11, p. P11010, 2006.

[27] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," Journal of statistical mechanics: theory and experiment, vol. 2008, no. 10, p. P10008, 2008.

[28] J. Liu and T. Liu, "Detecting community structure in complex networks using simulated annealing with k-means algorithms," Physica A: Statistical Mechanics and its Applications, vol. 389, no. 11, pp. 2300–2309, 2010.

[29] R. Guimera and L. A. N. Amaral, "Functional cartography of complex metabolic networks," nature, vol. 433, no. 7028, p. 895, 2005.

[30] T. N. Bui and B. R. Moon, "Genetic algorithm and graph partitioning," IEEE Transactions on computers, vol. 45, no. 7, pp. 841–855, 1996.

[31] M. Tasgin, A. Herdagdelen, and H. Bingol, "Community detection in complex networks using genetic algorithms," arXiv preprint arXiv:0711.0491, 2007.

[32] C. Pizzuti, "Ga-net: A genetic algorithm for community detection in socialnetworks,"in International conference on parallel problem solving from nature. Springer, 2008, pp. 1081–1090.

[33] M. Gong, B. Fu, L. Jiao, and H. Du, "Memetic algorithm for community detection in networks," Physical Review E, vol. 84, no. 5, p. 056101, 2011.

[34] M. Gong, L. Ma, Q. Zhang, and L. Jiao, "Community detection in networks by using multiobjective evolutionary algorithm with decomposition," Physica A: Statistical Mechanics and its Applications, vol. 391, no. 15, pp. 4050–4060, 2012.

[35] C. Liu, J. Liu, and Z. Jiang, "A multiobjective evolutionary algorithm based on similarity for community detection from signed social networks," IEEE transactions on cybernetics, vol. 44, no. 12, pp. 2274– 2287, 2014.

[36] Y. Zeng and J. Liu, "Community detection from signed social networks using a multi-objective evolutionary algorithm," in Proceedings of the 18th Asia Pacific Symposium on Intelligent and Evolutionary Systems, Volume 1. Springer, 2015, pp. 259–270.

[37] H. Zhou, "Distance, dissimilarity index, and network community structure," Physical review e, vol. 67, no. 6, p. 061901, 2003.

[38] H. Zhou and R. Lipowsky, "Network brownian motion: A new method to measure vertex-vertex proximity and to identify communities and subcommunities," in International conference on computational science. Springer, 2004, pp. 1062–1069.

[39] P. Pons and M. Latapy, "Computing communities in large networks using random walks," in International symposium on computer and information sciences. Springer, 2005, pp. 284–293.

[40] F.-Y. Wu, "The potts model," Reviews of modern physics, vol. 54, no. 1, p. 235, 1982.

[41] M. Blatt, S. Wiseman, and E. Domany, "Superparamagnetic clustering of data," Physical review letters, vol. 76, no. 18, p. 3251, 1996.

[42] J. Reichardt and S. Bornholdt, "Statistical mechanics of community detection," Physical Review E, vol. 74, no. 1, p. 016110, 2006.

[43] X. F. Wang and G. Chen, "Synchronization in small-world dynamical networks," International Journal of Bifurcation and Chaos, vol. 12, no. 01, pp. 187–192, 2002.

[44] A. Pikovsky, M. Rosenblum, and J. Kurths, Synchronization: a universal concept in nonlinear sciences. Cambridge university press, 2003, vol. 12.

[45] A. Arenas, A. Díaz-Guilera, and C. J. Pérez-Vicente, "Synchronization reveals topological scales in complex networks," Physical review letters, vol. 96, no. 11, p. 114102, 2006.

[46] A. Pluchino, V. Latora, and A. Rapisarda, "Changing opinions in a changing world: A new perspective in sociophysics," International Journal of Modern Physics C, vol. 16, no. 04, pp. 515–531, 2005.

[47] S. Gregory, "An algorithm to find overlapping community structure in networks," in European Conference on Principles of Data Mining and Knowledge Discovery. Springer, 2007, pp. 91–102.

[48] S. Gregory, "A fast algorithm to find overlapping communities in networks," in Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Springer, 2008, pp. 408–423.

[49] J. Baumes, M. K. Goldberg, M. S. Krishnamoorthy, M. MagdonIsmail, and N. Preston, “Finding communities by clustering a graph into overlapping subgraphs.” IADIS AC, vol. 5, pp. 97–104, 2005.

[50] S. Kelley, “The existence and discovery of overlapping communities in large-scale networks,” Ph.D. dissertation, Rensselaer Polytechnic Institute, 2009.

[51] G. Palla, I. Derényi, I. Farkas, and T. Vicsek, “Uncovering the overlapping community structure of complex networks in nature and society,” nature, vol. 435, no. 7043, p. 814, 2005.

[52] J. M. Kumpula, Kivelä, K. Kaski, and J. Saramäki, “Sequential algorithm for fast clique percolation,” Physical Review E, vol. 78, no. 2, p. 026109, 2008.

[53] T. Chakraborty and A. Chakraborty, “Overcite: Finding overlapping communities in citation network,"in Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining. ACM, 2013, pp. 1124–1131.

[54] Y.-Y. Ahn, J. P. Bagrow, and S. Lehmann, “Link communities reveal multiscale complexity in networks,” nature, vol. 466, no. 7307, p. 761, 2010.

[55] Y. Kim and H. Jeong, “Map equation for link communities,” Physical Review E, vol. 84, no. 2, p. 026110, 2011.

[56] T. S. Evans, “Clique graphs and overlapping communities,” Journal of Statistical Mechanics: Theory and Experiment, vol. 2010, no. 12, p. P12037, 2010.

[57] M. Rosvall and C. T. Bergstrom, “Maps of random walks on complex networks reveal community structure,” Proceedings of the National Academy of Sciences, vol. 105, no. 4, pp. 1118–1123, 2008.

[58] U. N. Raghavan, R. Albert, and S. Kumara, “Near linear time algorithm to detect community structures in large-scale networks,” Physical review E, vol. 76, no. 3, p. 036106, 2007.

[59] S. Gregory, "Finding overlapping communities in networks by label propagation," New Journal of Physics, vol. 12, no. 10, p. 103018, 2010.

[60] J. Xie, B. K. Szymanski, and X. Liu, “Slpa: Uncovering overlapping communities in social networks via a speaker-listener interaction dynamic process,” in 2011 IEEE 11th International Conference on Data Mining Workshops. IEEE, 2011, pp. 344–349.

[61] T. Nepusz, A. Petróczi, L. Négyessy, and F. Bazsó, "Fuzzy communities and the concept of bridgeness in complex networks," Physical Review E, vol. 77, no. 1, p. 016107, 2008.

[62] S. Mankad and G. Michailidis, “Structural and functional discovery in dynamic networks with non-negative matrix factorization,” Physical Review E, vol. 88, no. 4, p. 042812, 2013.

[63] J. Yang and J. Leskovec, “Overlapping community detection at scale: a nonnegative matrix factorization approach,” in Proceedings of the sixth ACM international conference on Web search and data mining. ACM, 2013, pp. 587–596.

[64] R. A. Rossi, B. Gallagher, J. Neville, and K. Henderson, “Modeling dynamic behavior in large evolving graphs,” in Proceedings of the sixth ACM international conference on Web search and data mining. ACM, 2013, pp. 667–676.

[65] S. Zhang, R.-S. Wang, and X.-S. Zhang, “Uncovering fuzzy community structure in complex networks,” Physical Review E, vol. 76, no. 4, p. 046103, 2007.

[66] M. Zarei, D. Izadi, and K. A. Samani, “Detecting overlapping community structure of networks based on vertex–vertex correlations,” Journal of Statistical Mechanics: Theory and Experiment, vol. 2009, no. 11, p. P11013, 2009.

[67] I. Psorakis, S. Roberts, M. Ebden, and B. Sheldon, “Overlapping community detection using bayesian non-negative matrix factorization,” Physical Review E, vol. 83, no. 6, p. 066114, 2011.

[68] L. Zhang, H. Pan, Y. Su, X. Zhang, and Y. Niu, “A mixed representationbased multiobjective evolutionary algorithm for overlapping community detection,” IEEE Transactions on Cybernetics, vol. 47, no. 9, pp. 2703– 2716, 2017.

[69] N. Srinivas and K. Deb, “Muiltiobjective optimization using nondominated sorting in genetic algorithms,” Evolutionary computation, vol. 2, no. 3, pp. 221–248, 1994.

[70] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: Nsga-ii,” IEEE transactions on evolutionary computation, vol. 6, no. 2, pp. 182–197, 2002.

[71] K. Deb and H. Jain, “An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: solving problems with box constraints,” IEEE Transactions on Evolutionary Computation, vol. 18, no. 4, pp. 577–601, 2014.

[72] J. Han, M. Kamber, and J. Pei, “Data mining concepts and techniques third edition,” Morgan Kaufmann, 2011.

[73] W. W. Zachary, “An information flow model for conflict and fission in small groups,” Journal of anthropological research, vol. 33, no. 4, pp. 452–473, 1977.

[74] D. Lusseau, "The emergent properties of a dolphin social network," Proceedings of the Royal Society of London B: Biological Sciences, vol. 270, no. Suppl 2, pp. S186–S188, 2003.

[75] M. Atzmueller, S. Doerfel, and F. Mitzlaff, "Description-oriented community detection using exhaustive subgroup discovery," Information Sciences, vol. 329, pp. 965–984, 2016.

[76] V. Nicosia, G. Mangioni, V. Carchiolo, and M. Malgeri, "Extending the definition of modularity to directed graphs with overlapping communities," Journal of Statistical Mechanics: Theory and Experiment, vol. 2009, no. 03, p. P03024, 2009.

[77] A. Lancichinetti, S. Fortunato, and J. Kertész, "Detecting the overlapping and hierarchical community structure in complex networks," New Journal of Physics, vol. 11, no. 3, p. 033015, 2009.

[78] Z. Pawlak, "Rough sets and intelligent data analysis," Information sciences, vol. 147, no. 1-4, pp. 1–12, 2002.

[79] G. Wang, T. Li, J. W. Grzymala-Busse, D. Miao, and Y. Y. Yao, Rough Sets and Knowledge Technology: Third International Conference, RSKT 2008, Chengdu, China, May 17-19, 2008, Proceedings. Springer, 2008, vol. 5009.

[80] G.-y. Wang, "Rough set theory and knowledge acquisition," Xi'an Jiaotong University Press, Xi'an, vol. 1, no. 0, 2001.

[81] G. Wang, D. Miao, W. Wu, and J. Liang, "Uncertain knowledge representation and processing based on roughset," Journal of Chongqing University of Posts and Telecommunications (Natural Science Edition), vol. 22, no. 5, pp. 541–544, 2010.

[82] A. Jakalan, J. Gong, Q. Su, and H. Hu, "Community detection in large-scale ip networks by observing traffic at network boundary," in Proceedings of the World Congress on Engineering and Computer Science, San Francisco, CA, USA, 2015, pp. 19–21.

[83] H.-K.LiuandT.Zhou,"Empirical study of chinese city airline network," 2007.

[84] Z. Yue and C. S. Du Wen, "Application of complex network theory to urban transportation network analysis," Urban Transport of China, vol. 7, no. 1, pp. 57–65, 2009.

[85] R. Kinney, P. Crucitti, R. Albert, and V. Latora, "Modeling cascading failures in the north american power grid," The European Physical Journal B-Condensed Matter and Complex Systems, vol. 46, no. 1, pp. 101–107, 2005.

[86] S. Wasserman and K. Faust, Social network analysis: Methods and applications. Cambridge university press, 1994, vol. 8.

[87] C. Pizzuti and S. E. Rombo, "Algorithms and tools for protein–protein interaction networks clustering, with a special focus on population-based stochastic methods," Bioinformatics, vol. 30, no. 10, pp. 1343–1352, 2014.

[88] Y. Sun, R. Barber, M. Gupta, C. C. Aggarwal, and J. Han, "Co-author relationship prediction in heterogeneous bibliographic networks," in 2011 International Conference on Advances in Social Networks Analysis and Mining. IEEE, 2011, pp. 121–128.

[89] Q. Chen, T.-T. Wu, and M. Fang, "Detecting local community structures in complex networks based on local degree central nodes," Physica A: Statistical Mechanics and its Applications, vol. 392, no. 3, pp. 529–537, 2013

[90] Y. Li, Y. Wang, J. Chen, L. Jiao, and R. Shang, "Overlapping community detection through an improved multi-objective quantum-behaved particle swarm optimization," Journal of Heuristics, vol. 21, no. 4, pp. 549–575, 2015

[91] X. Wen, W.-N. Chen, Y. Lin, T. Gu, H. Zhang, Y. Li, Y. Yin, and J. Zhang, "A maximal clique based multiobjective evolutionary algorithm for overlapping community detection," IEEE Transactions on Evolutionary Computation, vol. 21, no. 3, pp. 363–377, 2017.

[92] S. Boccaletti, M. Ivanchenko, V. Latora, A. Pluchino, and A. Rapisarda, "Detecting complex network modularity by dynamical clustering," Physical Review E, vol. 75, no. 4, p. 045102, 2007.

[93] J. Reichardt and S. Bornholdt, "Detecting fuzzy community structures in complex networks with a potts model," Physical Review Letters, vol. 93, no. 21, p. 218701, 2004.

[94] C. K. Williams and M. Seeger, "Using the Nyström method to speed up kernel machines," in Advances in neural information processing systems, 2001, pp. 682–688.

[95] S. Boettcher and A. G. Percus, "Optimization with extremal dynamics," complexity, vol. 8, no. 2, pp. 57–62, 2002.

[96] P. Bak, C. Tang, and K. Wiesenfeld, "Self-organized criticality: An explanation of the 1/f noise," Physical review letters, vol. 59, no. 4, p. 381, 1987.

[97] L. Li, M. Du, G. Liu, X. Hu, G. Wu, Extremal optimization-based semi-supervised algorithm with conflict pairwise constraints for community detection, in: Advances in Social Networks Analysis and Mining (ASONAM), 2014 IEEE/ACM International Conference on, IEEE, 2014, pp. 180–187.

[98] K. Tamura and S. Miura, "Necessary and sufficient conditions for local and global nondominated solutions in decision problems with multiobjectives," Journal of Optimization Theory and Applications, vol. 28, no. 4, pp. 501–523, 1979.