2017

The Role of Prior Knowledge in Multi-Population Cultural Algorithms for Community Detection in Dynamic Social Networks

Mukund Pandey
University of Windsor

The Role of Prior Knowledge in Multi-Population Cultural Algorithms for Community Detection in Dynamic Social Networks

By

Mukund Pandey

A Thesis

Submitted to the Faculty of Graduate Studies

Through the School of Computer Science

in Partial Fulfillment of the Requirements for

the Degree of Master of Science

at the University of Windsor

Windsor, Ontario, Canada

The Role of Prior Knowledge in Multi-Population Cultural Algorithms for Community Detection in Dynamic Social Networks

By:

Mukund Pandey

APPROVED BY

S. Nkurunziza
Department of Mathematics and Statistics

M. Kargar
School of Computer Science

Z. Kobti, Advisor
School of Computer Science

May 08, 2017

DECLARATION OF CO-AUTHORSHIP / PREVIOUS PUBLICATION

I. Co-Authorship

I hereby declare that this thesis incorporates material that is result of joint research, as follows:

In Chapter 3, the concept of prior knowledge was introduced by me under the guidance of professor Z. Kobti. Some parts of Chapter 4 of the thesis were co-authored with Pooya under the guidance of professor Z. Kobti. In this chapter, the first author designed the concept of training environment and community detection process. To analyze the role of prior knowledge, the four different approaches introduced in this chapter were brainstormed by Pooya and me. Moreover, the concept of non-similar networks was given by me. In Chapter 5, the simulation experiments were performed by the first author. Data analysis and interpretation of the data was a group effort of Pooya and me, both. Furthermore, the statistical analysis and the graphing of results was done by the first author and me. Professor Z. Kobti provide feedback on refinement of idea of non-similarity and editing of the manuscript. Part of Chapter 6 is again a joint effort of both, me and Pooya where we find the role of prior knowledge.

I am aware of the University of Windsor Senate Policy on Authorship and I certify that I have properly acknowledged the contribution of other researchers to my thesis, and have obtained written permission from each of the co-author(s) to include the above material(s) in my thesis.

I certify that, with the above qualification, this thesis, and the research to which it refers, is the product of my own work.

II. Previous Publication

This thesis includes one original paper that has been previously published/submitted for publication in peer reviewed journals, as follows:

Thesis Chapter	Publication title/full citation	Publication status
Chapter [3], [4], [5], [6]	Zadeh, Pooya Moradian, Mukund Pandey, and Ziad Kobti. "A study on population adaptation in social networks based on knowledge migration in cultural algorithm." In Evolutionary Computation (CEC), 2016 IEEE Congress on, pp. 4405-4412. IEEE, 2016.	Published

I certify that I have obtained a written permission from the copyright owner(s) to include the above published material(s) in my thesis. I certify that the above material describes work completed during my registration as a graduate student at the University of Windsor.

III. General

I declare that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained a written permission from the copyright owner(s) to include such material(s) in my thesis.

I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee and the Graduate Studies office, and that this thesis has not been submitted for a higher degree to any other University or Institution.

ABSTRACT

The relationship between a community and the knowledge that it encompasses is a fundamentally important aspect of any social network. Communities, with some level of similarity, implicitly tend to have some level of similarity in their knowledge as well. This work does the analysis on the role of prior knowledge in Multi-Population Cultural Algorithm (MPCA) for community detection in dynamic social networks.

MPCA can be used to find the communities in a social network. The knowledge gained in this process is useful to analyze the communities in other social networks having some level of similarity. Our work assumes that knowledge is an integral part of any community of a social network and plays a very important role in its evolution. Different types of networks with levels of non-similarity are analyzed to see the role of prior knowledge while finding communities in them.

DEDICATION

This thesis is dedicated to God, my beloved parents, my sister and my brother-in-law who have always been there for me and supported me during my all ups and downs. Their love and motivation has kept me focused this far.

I cannot forget my cutest nephew, Archit, who always made me smile whenever I think about him.

Also, this thesis is dedicated to one of my friends Pooya who has guided me through every step of this journey.

ACKNOWLEDGEMENTS

I would like to thank my supervisor, Dr. Ziad Kobti, for his continuous assistance and support during my entire Master’s study. I would like to thank him for believing in me and motivating me during my thesis work.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ACRONYMS & DEFINITIONS

EC- Evolutionary Computation

EA- Evolutionary Algorithm

GA- Genetic Algorithm

CA- Cultural Algorithm

MPCA- Multi-Population Cultural Algorithm

TAMPCA- Transfer-Agent based Multi-Population Cultural Algorithm

OSLOM- Order Static Local Optimization Method

MCAKM- Multi-population Cultural Algorithm adopting Knowledge Migration

Candidate Solution- Individual

Population- Group of Individuals

Search Space/ Environment- Problem

CHAPTER 1
INTRODUCTION

1.1 OVERVIEW

At the advent of computers and the early stages of the internet, the data was transferred only in the form of simple text files and some very low-end graphics. The initial concept of the Internet (Web 1.0) was to communicate with a limited number of computing devices. Users were supposed to do the passive viewing of contents posted on the websites. As the time passed and the technology fostered, the web also got more advanced, and Web 2.0 came into the picture. The users of the Web 2.0 could generate their content, and to interact and collaborate with each other. The content on the internet, in contrast to Web 1.0, were linked to each other. Users could go from one page to another just by clicking on the links provided on the same website or a different one. That was the starting of social networking websites and social media.

In the last few decades, as the World Wide Web has expanded its reach among people, the network of social interactions and personal relationships has grown rapidly. There are new online users coming every single day, and, so far nearly one-fourth of the entire population of the world has come in touch with social networks [29]. An online social network can be described as a dedicated website or other application that helps used to communicate with each other in the form of texts, images, videos, etc. In other words, we can say that social networks are a collection of nodes and their linking edges where the nodes are people, and the links imply the social relationship between them. A social relationship could be anything like beliefs, thoughts, customs, language, friendship, ethnicity, language, etc. that two people share with each other. A dynamic social network, as the name suggests, is active in nature and changes at the time. New nodes might join the network or old nodes may leave the network. Similarly, two nodes might change the social relationship between them and so the connection between them, as the time passes.

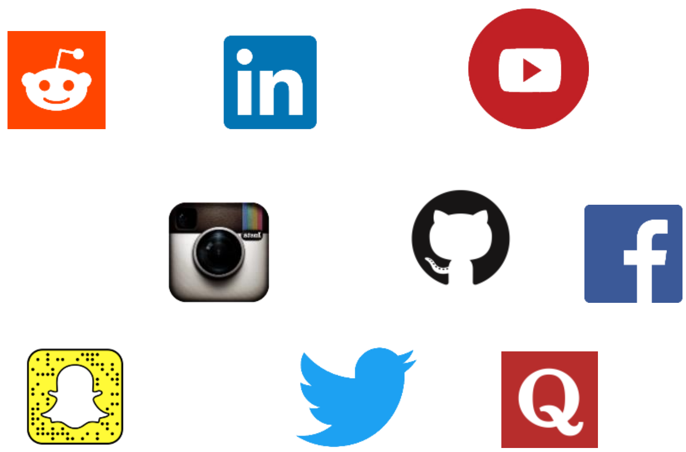

Figure 1.1: An Example of Some Social Networks

A group of nodes that possess same "liking" in their beliefs, friendship, thoughts, geography, or language, etc. predominantly have a better communication with each other. These nodes usually associate with each other more closely and form a community inside a social network. The interaction between nodes in a community is higher as compared to the nodes which don't belong to this community. In other words, we can say that community in a network is a group of nodes within which the network is densely connected but between which the network is sparsely connected [1]. Analyzing the networks and detecting their communities is one of the prime concentration areas among the computer scientists. The reason behind is, networks and their communities explain a lot about their surrounding environments. Having some knowledge about their behavior and characteristics can reveal a lot of unknown patterns about their environment and characteristics [1][2].

In social networks, nodes are called social agents. These social agents are a socially-active entity which transmits the knowledge to the other agents and at the same time they generate and deliver new content into the network as well. So, parallelly, nodes are considered the source of new knowledge. Moreover, agents in social networks own some knowledge about their neighbors and their environment. Evolution of knowledge is, in general, the most important driving force in the evolution of a community and finally the whole social network.

If there is some knowledge about the network and its communities regarding structure, nodes, and social agents, it is possible to understand the real-world societies and individuals [1][2] and the important features of their environment more extensively. Because of the wide range of its implementation, ranging from understanding the biological and social behavior of individuals, advertising, finances, human-welfare, etc., community detection is very important in today's scenario.

There are many different community detection algorithms devised in the field of computer science to detect communities in the social networks. They range from finding communities in static to dynamic networks and from small to large networks, from finding communities which are known in advance in a network to the ones where nothing is known about the communities in a network. The most important among all of them are Evolutionary Algorithms (EA). These are the subset of evolutionary computation and are based on the mechanisms inspired by biological evolution such as reproduction, mutation, recombination, and selection. One type of evolutionary algorithms which is best known for solving the optimization problems is called Genetic Algorithm (GA). GA is one of the most commonly known types of evolutionary algorithms, and it works on the concept of evolution among living beings which is predominated by their genes or in other words, the genetic evolution.

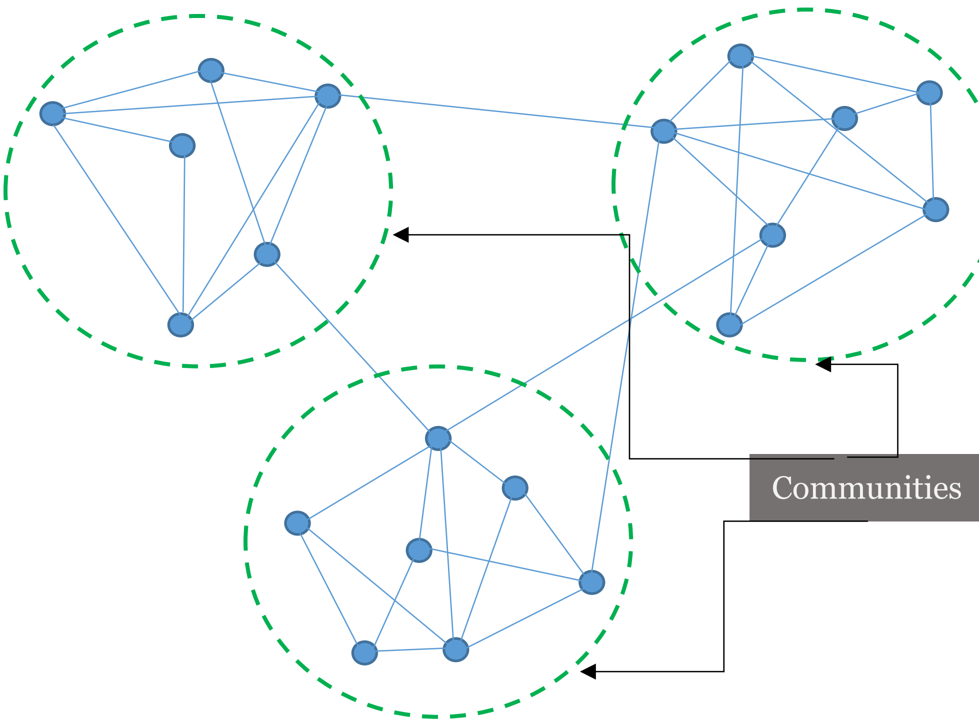

Figure 1.2: Communities in a Networks

The Cultural algorithm is an extension to the EA and GA. In GA, the population is supposed to evolve only with its genes over a period of generations. Genes are the only driving force that can evolve any population. Contrast to this, in the cultural algorithms there are two driving forces that help evolve any population- the population itself and the knowledge that this population contains. This knowledge is also known as belief space. The belief space has different categories that represent different domains of knowledge that a population persists. In CA, the knowledge that any population contains plays a very important role in its evolution [3].

A network can be defined as a graph G (V, E) where V= $\{v_1,v_2,\ldots v_n\}$ is the set of objects called nodes, and E= $\{(v_i,v_j),\ldots\}$, where vi,vj belong to V, is the set of links called edges that connect two elements of V. This graph can be represented in the form of an adjacency matrix to define a social network in a computer. A graph with N nodes can be represented using an N X N adjacency matrix. A population is a set of solutions called individuals that a graph can have. The best individuals and the knowledge of any population are responsible for its evolution. Best individuals are chosen with the help of fitness functions.

1.2 MOTIVATION

In the recent years, social networks have expanded many folds, and its reach has covered almost half of our population. Due to that, network analysis is in the center of attraction for a vast majority of researchers. Communities are one integral part of these social networks, and social networks can be characterized as a group of one or more communities. Understanding these communities can give a good insight about its constituents, and it can help explore some important information about its environment. Assuming knowledge is the distinguishing factor of any social network, and it plays an important role in its evolution, if there is a change in the network's knowledge there should be an impact on its evolution.

If the knowledge of any population migrates, it could affect the evolution of the network where it goes. In other words, we can say that migration of knowledge can affect the evolution of any social network. To study this impact of prior knowledge on community detection using Multi-Population Cultural Algorithm (MPCA) in social networks is the target of this research.

1.3 PROBLEM STATEMENT

Through the work done in the past in the field of Cultural Algorithm to solve the problem of community detection in social networks, it can be said that knowledge obtained by any network is very important for its evolution [3]. In this research, we are interested in determining the role of prior knowledge in enhancing the community detection problem in social networks. Particularly, the target is to know what is the role of a population's prior knowledge when it migrates from one network to another network. The goal is to find out how a migrated population affects the community-detection process when the knowledge which it has gained is also transferred. To analyze this, MPCA has been used to solve the community detection problem taking different scenarios of prior knowledge migration from one population to another.

1.4 ORGANIZATION OF THESIS

This thesis addresses the role of prior knowledge in MPCA for community detection in dynamic social networks. This thesis contains six chapters. Chapter two of this thesis contains a brief survey of related topics that have included the approaches of community detection in social networks and the background review of knowledge sharing/migration. Chapter three gives a detailed overview of the MPCA and what prior knowledge is. Chapter four covers the implementation details of the proposed analysis such as the representation of networks, individuals, and other useful methodologies used afterward. It also includes the brief information about different scenarios that we are going to test. Chapter five provides a comprehensive analysis of the results that have been gathered using different approaches and its comparison. Chapter six contains conclusions and future work.

CHAPTER 2

BACKGROUND INFORMATION

This chapter describes the fundamentals of social networks and the importance of community detection. After that, it explains some of the most useful algorithms for community detection in networks. After that, this chapter briefly illustrates the relevant literature review done on community detection problem in social networks. It also provides the necessary overview of the literature on knowledge sharing and migration. The related works segment explains some related work done in the past.

2.1 FUNDAMENTALS OF SOCIAL NETWORKS

Networks are a system of interconnections where the elementary units of the system are called nodes or vertices, and the connections that connect these nodes are called links or edges. It is easy to find the two main constituents of a network- nodes and links and therefore it's every common to find networks in many different aspects of life such as, in biology, where proteins interact with each other, in ecology, where species interact to their nutrition connection, society, where people interact with others, in the internet where routers and computers connect to each other with or without wires [4].

Social networks are one the most important networks in today's world where people from around the globe join the same platform to communicate with others. People may live on the other side of the earth, but they can interact with their friends and family. All of this is possible with the help of social networks. Social networks are the collection of nodes which represent people and links that commonly represent social relations. Some social relations could be common interests, culture, language, etc. A social network shows the relationship between people and the flow of knowledge between them.

2.2 COMMUNITY DETECTION

Communities are the group of nodes in a network that is densely connected to each other within the network and is sparsely connected to the other nodes present in the whole network. Nodes with similar properties and functions are likely to connect to each other rather than connecting with nodes that have no commonalities. These nodes form a highly cohesive subgroup within the network called community. One interesting property to investigate in any network is its communities. Communities reveal a lot about the networks and their individuals. Community and its structure show how the connections between the individuals affect the individuals and their relationships within the network. Also, finding communities can help to identify functional subunits of a network and the similarities among individuals that are normally hidden under the networks. The community structure is also a very useful way to represent a network and its connections where visualizing each node is not possible [4] [5].

2.3 EVOLUTIONARY ALGORITHM, GENETIC ALGORITHM, AND CULTURAL ALGORITHM

Many approaches have been proposed from different fields of computer science, mathematics, biology to solve the optimization problems. One such family of algorithms, inspired by biological evolution, proposed to solve the problem of global optimization is known as Evolutionary Computation. Evolutionary Computation is a population-based trial-and-error problem-solving algorithm. The one characteristic that makes them a biologically inspired algorithm is their stochastic optimization character which chooses random solution set for the problem and then randomly updates the solutions set by removing the less desired results and adding new changes.

An evolutionary algorithm is a subset of evolutionary computation, which is based on the generic population-based optimization algorithm. An evolutionary algorithm is inspired by the biological evolution such as mutation, reproduction, recombination, and selection. In EA, to solve the problem of optimization, the initial set of solutions is generated randomly. A new population is then iteratively generated by updating the previous population and by calculating its fitness value. The less fit solutions of the previous population are removed, and the ones which have more fitness value take their place. Parallel to this process, a small number of new random individuals are also introduced to the population. Gradually, the population evolves as its fitness increases. Usually, a fitness function is used to determine the fitness of the population, and the evolution process stops as the required fitness value is achieved.

A genetic algorithm is a type of evolutionary algorithm. Like EA, GA is a metaheuristic inspired by the process of natural selection in nature where the process of evolution takes place in every new population. Genetic algorithms help to solve the complex optimization problems using the nature inspired processes such as mutation, crossover, and selection. In GA, the set of candidate solutions are known as individuals and the who set is called a population. The individuals are evolved towards the better solutions while working on the optimization problem until the result is achieved. The evolution process is started using a population of randomly generated candidate solutions i.e. individuals. These populations are updated in every iteration which is known as a generation. The fitness value of every individual in calculated in each generation and the individuals with higher fitness value are selected from the current population. Then a new generation if generated after the genome of each present in the population is changed using the recombination and mutation process. This new generation is used to generate the new generation of more fit individuals. This iterative process can happen until the end condition is met. By this process, GA produces high-quality solutions to the optimization problems.

One more type of algorithm is introduced to solve the optimization problems is known as Cultural Algorithm. CA is an extension to the evolutionary and genetic algorithm with an extra space called knowledge or belief space in addition to the population space present in the evolutionary and genetic algorithm. The rationale behind the advent of CA is the belief that populations don't evolve only with the genetic mutation and crossover over the time. One more aspect that affects the evolution of each generation is the knowledge that it inherits from its previous generations. This gained knowledge helps the generation to evolve faster than the genetic mutation which takes many generations to occur and make some visible changes. In CA, every population helps to figure out the best individuals among them using their fitness value, and then these best individuals are used to update the knowledge/ belief space of the whole population. This new belief space helps to generate the new population in the next iteration. This new generation introduced with the help of belief space evolves quickly with better fitness.

2.4 LITERATURE REVIEW

2.4.1 COMMUNITY DETECTION

In recent years, a lot of work has been intensively done in the field of complex networks for detecting communities among them [1] [4] [5] [6].

Newman et al. [7] are keen to address the problem of community detection in networks. According to them, community detection is one of the main factors to understand any network. As per their research, many of the previously suggested methods fail with some frequency in the case where community structure is already known. Another drawback of some of these methods (agglomerative) is that they find only the tightly connected nodes of any community and leave its less densely connected nodes i.e. they are not able to find whole communities efficiently. In this work, the authors propose a class of new divisive algorithms for community detection in networks. The work is done on non-dynamic networks where there are no common vertices between communities; edges are undirected and with no weight. The algorithm presented here find edges which are connected to highly connected nodes. This is termed as vertices with the highest “betweenness.” One of the algorithms calculates betweenness with the help of shortest path. According to the second method, the shortest-path betweenness is thought as a signal traveling through the network. The betweenness in this method is calculated as the rate at which signals pass along each edge. The general form of proposed community detection algorithm is. First, the authors calculate betweenness of all the edges in the network. Then they find the edge with the highest score and remove it from the network. After that, they recalculate betweenness of all the remaining edges in the network. Finally, the algorithm repeats itself from the second step to find a community.

For their experiments, Newman et al. [7] used some known networks with known community structure to test their algorithm. Three real networks are Zachary’s karate club network where a total number of nodes were 32 and two communities, Collaborative network with 13 communities was tested, and the network of 62 dolphins. Furthermore, under the computer-generated networks, many graphs were generated with 128 nodes and then these graphs were divided into four communities with 32 nodes in each.

The authors claim that they analyzed the graphs for understanding the performance of proposed algorithm. For computer-generated networks, the authors claim to have achieved more than 90% performance regarding finding communities for both shortest-path and random walk methods for all vertices classified correctly from Z_{out}= 0 to Z_{out}= 6. For Z_{out} >= 6, the authors felt some classification deterioration.

The authors believe that the proposed algorithm is a good solution for community detection in networks. The authors claim that the introduction of a "divisive" technique which iteratively removes edges from the network and the use of recalculation step introduced helped to identify communities in networks.

In the work of Lin et al. [8], they propose a systematic framework FacetNet for analyzing communities and their temporal evolution in complex networks. As per the authors, a unified framework is more suitable for analyzing communities and their evolution where there is a high variation in the community structure while it evolves. This kind of framework is more appropriate to analyze which community structure is more suitable with the help of evolutionary history. Evolution of each community is monitored at each timestamp t using the network data and the historic community evolution pattern using the FacetNet framework. Also, they extend the soft clustering algorithm by making the communities overlapping to each other at different levels. Furthermore, the authors provide a local optimal based community detection algorithm in dynamic networks.

Lancichinetti et al. [4] propose Order Statics Local Optimization Method (OSLOM) to overcome issues related to dynamic and overlapping networks. OSLOM model works on the statistical significance of clusters for community detection. OSLOM uses the fitness function to evaluate the clusters for statistical significance. Networks are configured randomly by configuring random nodes keeping in mind that every node has fixed number of vertices from the per-assigned degrees. OSLOM consists of three phases- first, it searches for significant clusters until convergence; in the second phase, it tries to detect the internal structure of the clusters found in the previous step; in the third step detection of the hierarchical structure of clusters is done. The main characteristic of OSLOM is that it is based on fitness score, which is very much related to the significance of clusters in the configuration model.

The work done by Pasquale et al. [9] concentrates on community detection in large networks. It depends on the well-known concept of network modularity optimization. For optimization, the proposed algorithm find the edge centrality to rank the edges of a large network. After that, the authors calculate the pairwise node proximity in the network. The community structure is then figured out with the help of Louvain method, maximizing the modularity of the network.

A Cultural algorithm which is a branch of the evolutionary algorithm has an extra edge over the other algorithms with the introduction of a knowledge space in it. Knowledge (or belief space) helps the process to converge faster than other evolution based algorithms. Zadeh et al. [10] propose a Multi-Population Cultural Algorithm (MPCA) for community detection in social networks. As per the authors, the new proposed algorithm finds communities faster and more accurately, in most of the cases, than other well-known methods in the field.

2.4.2 KNOWLEDGE MIGRATION

Knowledge component is very important in cultural algorithms. It helps evolve the network faster and more accurately. This knowledge could be transferred among different populations so they can learn about the environment where the knowledge comes from. As per our knowledge, there is not much work done in the field of knowledge migration among networks.

Guo et al. [11] on the other hand propose a novel MPCA which accepts the knowledge migration among different populations with the help of individual migration. This deals with the problem of not to have a substantial evidence of evolution when an individual migrates to a different population, which limits the optimization performance. Authors introduce Multi-Population Cultural Algorithm adopting Knowledge Migration (MCAKM) to solve the problem of individual migration with less evolutionary knowledge about its population. It helps to exchange the implicit knowledge among sub-populations.

In contrast, Hlynka et al. [12] have studied a new method of knowledge transfer using agent migration in MPCA- Transfer-Agent based MPCA i.e. TAMPCA. The authors claim that the effect of individual transfer between two subpopulations depends on the percentage of individuals transferred and the quality of the belief space. For their work Hlynka et al. [12] have used two different types of knowledge for two sub-populations- topographic and situational knowledge.

CHAPTER 3

MPCA- A DETAILED OVERVIEW

The aim of this chapter is to provide the reader a detailed overview of the cultural algorithm and its variant multi-population cultural algorithm. This chapter explains different functionalities and methodologies used to define an MPCA. Furthermore, different types of knowledge and their implementation are explained in detail. Some other concepts which are coined for this research work have been discussed in brief in the end.

3.1 WHY MULTI-POPULATION CULTURAL ALGORITHM (MPCA)?

In dynamic social networks both time and space play a crucial role while detecting the communities inside it. Evolutionary computation and its extension algorithms opened a new window to analyze the optimization problems differently. Still, the implicit information contained in the population got wasted for global optimizations in the cases where knowledge and individual transfer takes place between different networks. The introduction of knowledge in addition to the population space gives cultural algorithms an extra advantage over other evolutionary computation algorithms. With the help of belief space, cultural algorithms can converge faster and with more accurate global optimal resulting in a comparatively more fit population. Furthermore, because the evolution process does not depend only on the genes of the population, and the knowledge is a very integral part of it, the evolution process of any population using CA is faster as compared to any evolutionary computation algorithm such as an evolutionary algorithm or genetic algorithms.

3.2 ALGORITHM

```
CULTURAL ALGORITHM
begin
     t=0;
     Initialize Population POP(0);
     Initialize Belief Network BLF(0);
     Initialize Communication Channel CHL(0);
     Evaluate (POP(0));
     t=1;
     repeat
          Communicate (POP(0), BLF(t));
          Adjust (BLF(t));
          Communicate (BLF(t), POP(t));
          Modulate Fitness (BLF(t), POP(t));
          t = t+1;
          Select POP(t) from POP(t-1);
          Evolve (POP(t));
          Evaluate (POP(t));
     until (termination condition)
end
```

This algorithmic view provides an overview of the metaheuristic cultural algorithm. Here, POP and BLF represent the population and belief space of any network respectively. A communication channel is a set of rules defined by a population and belief space through which desired individuals in the population can influence the belief space.

In optimization problems, a candidate solution is defined as a member of the set of possible solutions that fall in the feasible region of a solution of a given problem. These candidate solutions are also known as individuals. An individual is not always a likely solution or reasonable solution, but it satisfies all the minimum requirements that a solution should have. A collection of these individuals is defined as population. A population space is such a set of individuals that help to achieve the optimal solution of a problem.

Every population has some knowledge about the problem it predominantly wants to solve. The metaphor used to address a problem is the environment. So, every population has some knowledge about its environment. This implicit knowledge plays an important role in the cultural algorithm to optimize a problem. A belief space stores the best knowledge about its population's environment. This dual inheritance model of the cultural algorithm allows it to both learn and adapt the best features of any population [13].

Figure 3.1 shows the basic model of any cultural algorithm. In the starting, like in any other evolutionary computation algorithm, when time (t=0) a random set of individuals is generated and used as an initial population to initialize the population space POP(0). Similarly, a belief space is initialized at the initial time BLF(0). A fitness function is then used to filter out the best candidate solutions by calculating their fitness value. These selected fit individuals are among those individuals that can update the belief space. After this, based on some predefined criteria the accept function is used to allow some of the fit candidate solutions to update the belief space. In continuation to this, the knowledge is extracted from the best-fit-individuals, and the update function is called to renew the belief space. In the next step, a new population is generated with the help of rules defined in the influence function [3], [13], [10]- [14].

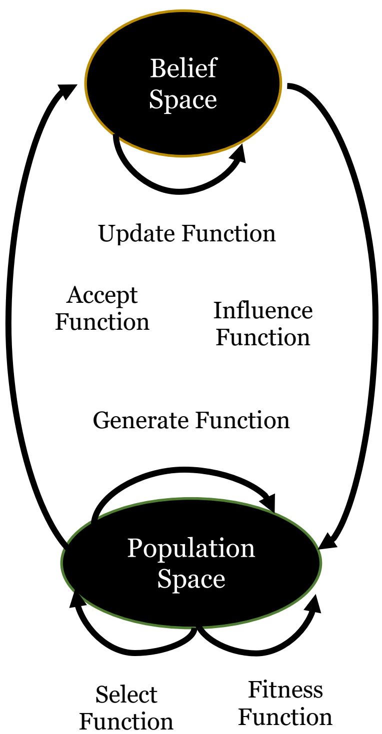

Figure 3.1: A Cultural Algorithm Mode

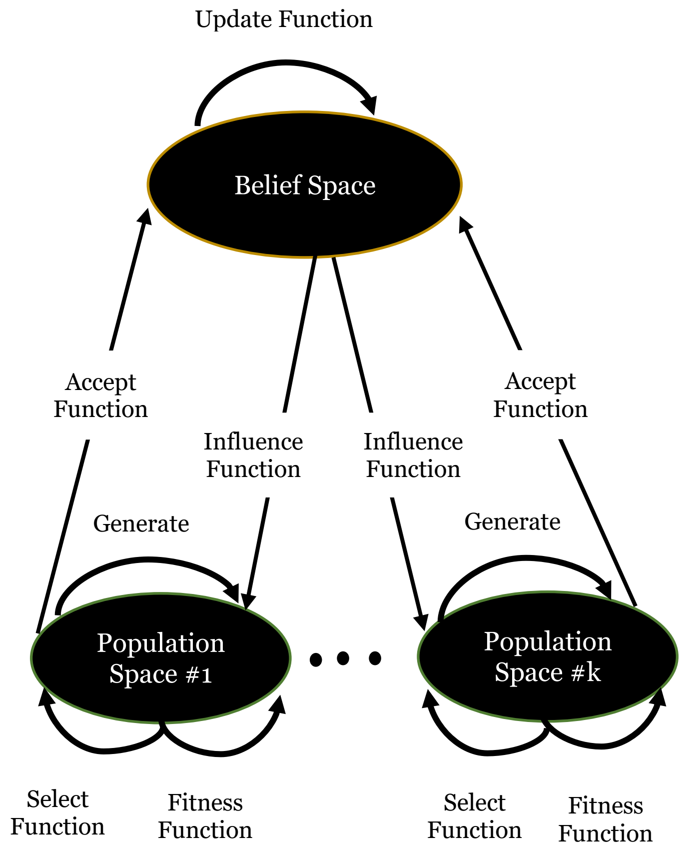

Figure 3.2: A Multi-Population Cultural Algorithm Model

Furthermore, to reduce the processing time and the complexity to solve the optimization problem, an extension of the cultural algorithm was introduced which is known as Multi-Population Cultural Algorithm [10]- [12], [14], [15]. Figure 3.2 shows a model of MPCA where a population is divided into subpopulations. All these sub-populations share a common belief space which helps all of them evolve parallelly.

3.3 CATEGORIES OF BELIEF SPACE

The most important part of cultural algorithms that differentiate them from other evolutionary algorithms is knowledge. Different types of knowledge are used to solve different types of optimization problems. A belief space could be of different categories depending on the type of knowledge that a population could have about its environment [10], [14], [16]. There are five different categories of knowledge that have been proposed [16] [17]-

- Normative Knowledge: The normative knowledge describes a range of acceptable solutions within which changes in the individual can still converge in the desired result for any optimization problem. Normative knowledge helps candidate solutions to adjust into a more desirable range if they are not already there.
- Situational Knowledge: The situational knowledge gives an illustrious set of cases that are useful for knowing any individual's experience. This kind of knowledge is used to solve real-valued optimization problems. Situational knowledge magnetizes the individuals towards the optimal solutions.
- Topographical Knowledge: This type of knowledge uses the whole search space and then divides it into smaller cells. It's easy for each cell to keep track of best individual in its domain. Topographical knowledge usually provides local optimal solutions by guiding its individuals towards the cell-best.
- Domain Knowledge: The domain knowledge guides a search by utilizing the knowledge about its problem domain.
- Historical or Temporal Knowledge: This kind of knowledge is used to analyze the search process and then keeps track of the important trends in the

search. These trends might be some important change in the search space, or it might refer to change in the landscape.

These five types of knowledge are added into the cultural algorithm by different researchers to solve different types of optimization problems [18]- [19].

3.4 PRIOR KNOWLEDGE

The five different types of knowledge we studied in the previous section cover all type of problem domains that can occur for optimization. Some combination of this five knowledge can completely express any cultural knowledge [17].

Prior knowledge can be defined as the knowledge that one population gets from a different population that got trained in an environment that is not starkly different from its environment i.e. there is some level of similarity between the two environments. The similarity between the two environments ensures that their solutions could be similar as well. Figure 3.3 shows the pictorial representation of prior knowledge.

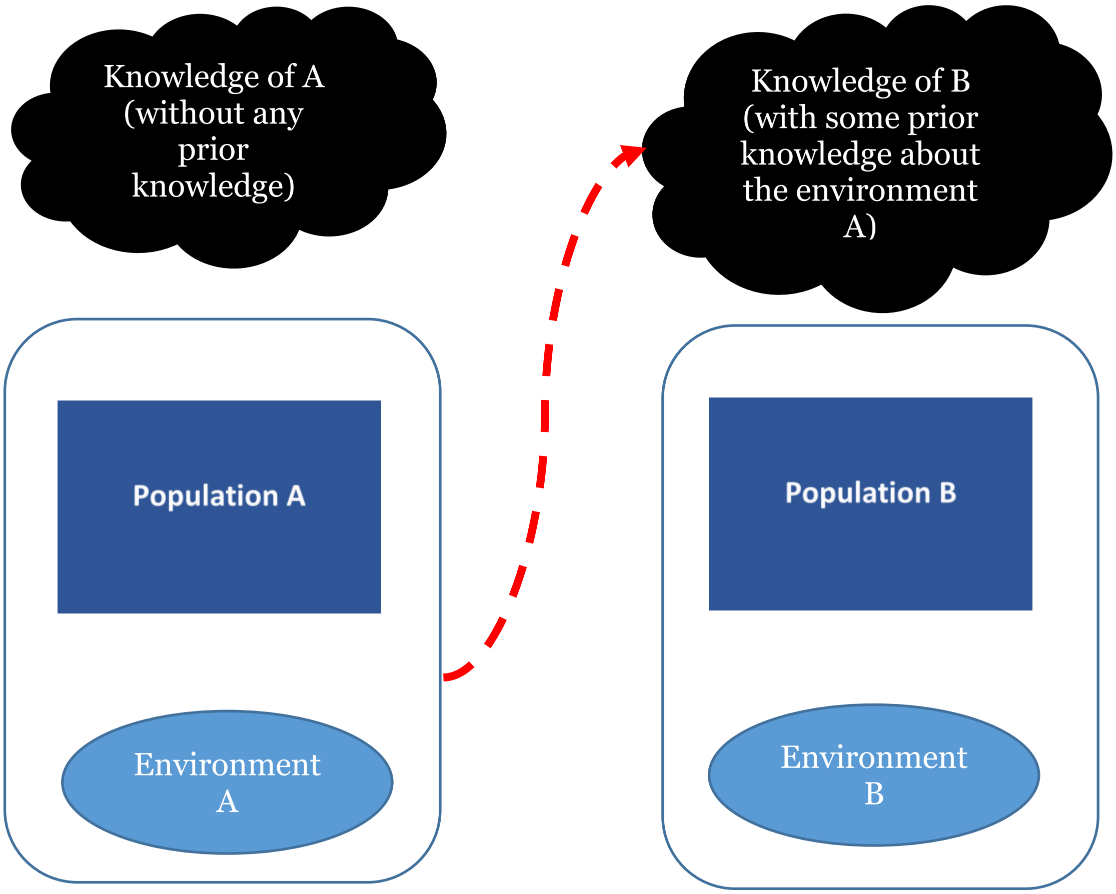

Figure 3.3: Prior Knowledge

When cultural algorithms are applied, a population living in an environment gets optimized after few generations (iterations). The knowledge gained during this process could be precious for new populations that have not started their optimization process yet. This transfer of knowledge from environment A to environment B is called prior knowledge. This kind of knowledge does not have to do with the five different categories of knowledge we have discussed. It is just the transfer of the gained knowledge from one population to another.

CHAPTER 4
DESIGN AND METHODOLOGY

This chapter is intended to introduce the design and methodology used for the multi-population cultural algorithm to understand the role of prior knowledge for community detection in dynamic social networks. First, an illustrative representation of network and individual is done. After that, in the next section, the fitness function and its role are explained. After that, the training environment and the non-similar networks are briefly presented. This section also explains different scenarios that are possible for analysis. The last part of this chapter explains the training process for the base network to get access to its knowledge for other non-similar networks.

4.1 NETWORK AND INDIVIDUAL REPRESENTATION

Representing a network on a computer is different from what humans understand from a pictorial representation of a network. Networks, in computer science, are represented in the form of graphs. In any social network consists of people and a link between a group of them in fashion. In a graph representation, the nodes denote people, and the links represent the connection between them. A network can be defined as graph G (V, E), where V= $\{v_1, v_2, v_3, \ldots., v_n\}$ is the set of nodes and E= $\{(v_i, v_j), \ldots\}$, where $\mathbf{v_i}, \mathbf{v_j} \in V$, is the set of edges that connect two elements of V.

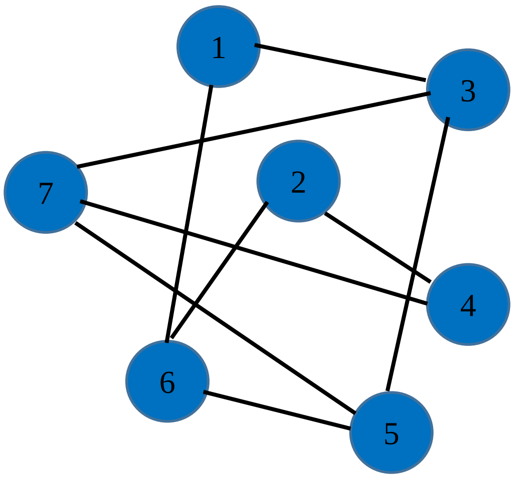

Figure 4.1: A sample graph

$$\begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \end{pmatrix}$$

Figure 4.2: Adjacency Matrix

Figure 4.1 and 4.2 represent how a graph for our research purpose is used. Undirected and unweighted graphs are used for this research work. These graphs are then represented in the form of an adjacency matrix. A graph G (V, E) with N nodes can be expressed using an N X N adjacency matrix A, with entry at position (i, j) as one if there is a link between the two nodes, and 0 otherwise [10] [11].

For community detection in the cultural algorithm, a candidate solution is represented by utilizing the adjacency matrix of the network [10]. An individual denotes an instance of the graph i.e. it is a random subset of the graph. This random subset is represented using locus-based adjacency representation of the matrix. This individual is represented as an array, and its size is equal to the number of nodes in the graph. Every index in this array is a direct mapping to the corresponding node in the network. Now, every index of this array is filled using one of the random nodes it is connected to in the network. For instance, in Figure 4.2, node one is connected to node 3 and 6, node two is connected to node four and6. Similarly, node 7 is connected to node 3, 4, and 5 [10] [14].

Figure 4.3: Three Random Individuals

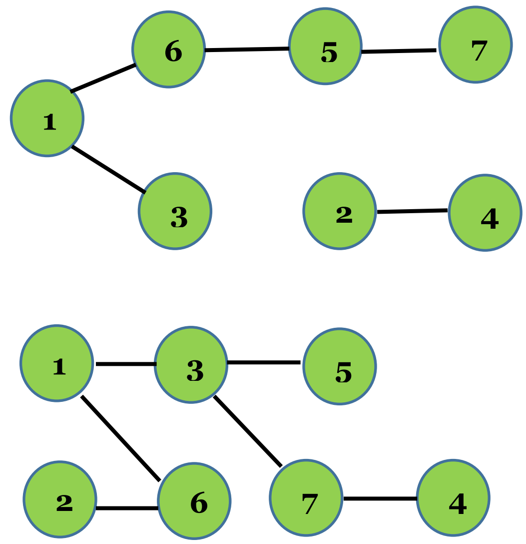

Figure 4.4: Decoding of individual 1 and 2

Based on the definition of an individual and using the mechanism described here, three random individuals are generated and shown in Figure 4.3. Every individual represents a probable community the forms the graph. After decoding an individual (Figure 4.4), all the links that are connected are considered as a subset of the network which is interpreted as a community. Different individuals split the network into different communities with random network nodes assigned to them.

4.2 FITNESS FUNCTION

A fitness function is an objective function that measures how close the candidate solution is to the desired result of the optimization problem. Furthermore, to discard the less optimal design solutions and select the ones that are more promising, it is important to analyze which individual falls under which category. For this purpose, many fitness functions have been proposed by different authors

In this proposed work, the concept of community score is used as a fitness function which is defined by [5].

For any sub-matrix S = (I, J) of adjacency matrix A, mean values of the i^{th} row and j^{th} column could be represented as-

$$a_{iJ} = \frac{\sum_{j\varepsilon J} aij}{|J|} \quad (1) \quad \text{And} \quad a_{Ij} = \frac{\sum_{i\varepsilon I} aij}{|I|} \quad (2)$$

Where, I am the subset of rows X= {I_1, I_2, …, I_n} of A, and

J is a subset of columns Y= {J_1, J_2, …, J_n} of A

Furthermore, volume Vs of sub-matrix S = (I, J) shows the number of 1 entries at a_{ij} such that i ∈ I and j ∈ J can be calculated as,

$$V_S = \sum_{i\varepsilon I, j\varepsilon J} a_{ij} \quad (3)$$

After calculating the number of 1s at a_{ij}, the score of each community can be calculated as,

$$Q(S) = \sum_{i \in I} \frac{\left(\frac{\sum_{j \in J} a_{ij}}{|J|}\right)^{\cdot}}{|I|} \times \sum_{i \in I,\, j \in J} a_{ij} \qquad (4)$$

And finally, the community score of k partitioning $\{S_1, \ldots, S_k\}$ of A can be defined as,

$$CS = \sum_{i=1}^{k} Q(Si) \qquad (5)$$

4.3 TRAINING ENVIRONMENT

As discussed in the previous sections, the knowledge gained from a population which has the knowledge about its environment could play some constructive role in a new population which needs to adapt to an environment similar to the previous population's environment.

To understand the role of prior knowledge it is important to generate a base network which can provide its knowledge for the further analysis in other networks. If the knowledge of a trained population is used as a base knowledge for a new population, it might be helpful for the new population to optimize the solution in time and space and adapt to the new environment. Community detection process described in [10] has been used in this work to train a network.

In the training process, a random network is designed as per the guidelines are given in [7] with random initial individuals and a community-detection process proposed in [10] is used to find out the network communities. The output of this process is the normative knowledge matrix, the best individuals and the number of iterations used to find out the communities. The algorithm stops when the number of predefined iterations has been achieved. Figure 4.5 shows the process of the training environment.

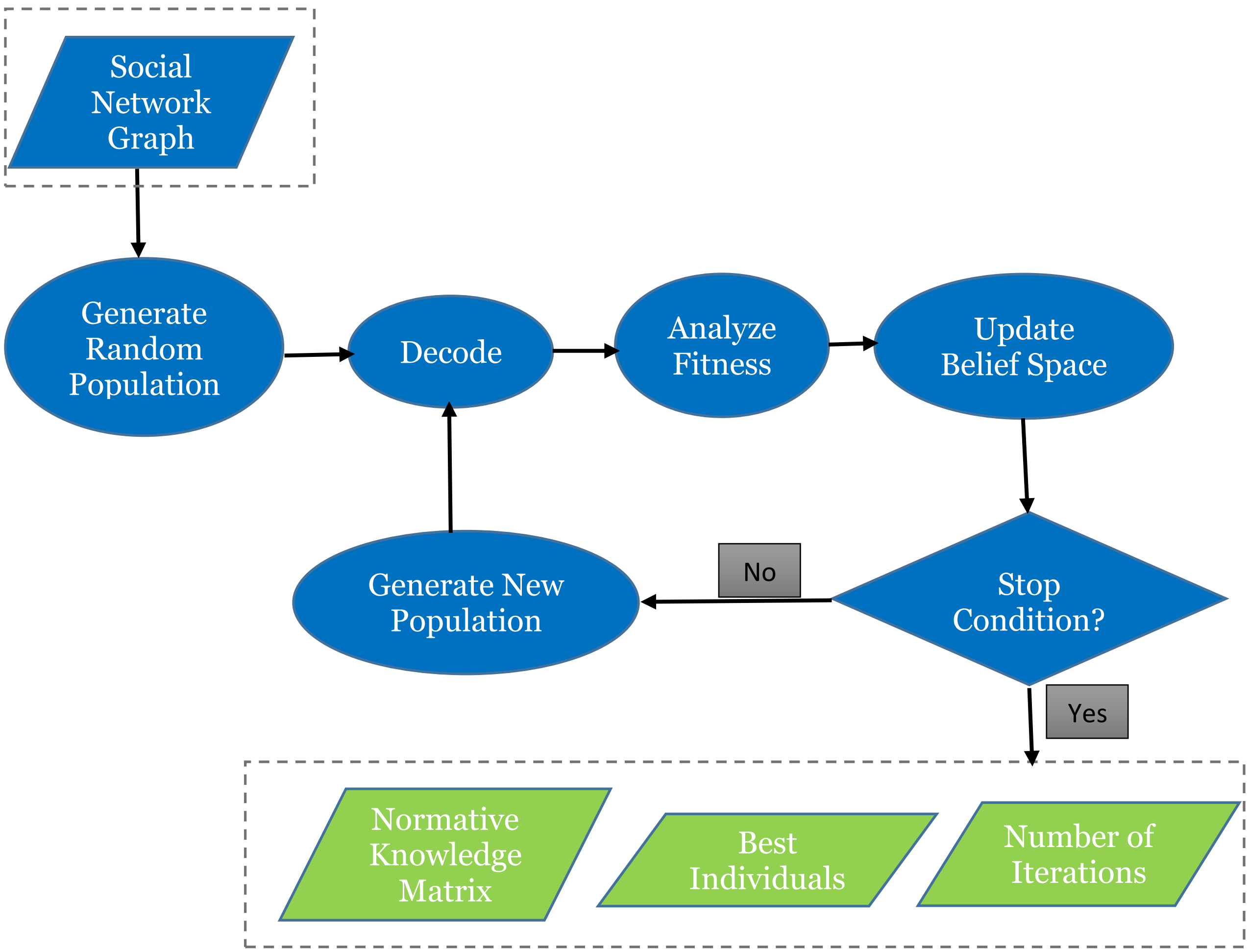

Figure 4.5: Training Environment

4.4 DIFFERENT APPROACHES

To implement the knowledge gained from the training environment new networks are created keeping in mind to have some level of non-similarity between the two networks. The new network is created from the previous network such that the number of nodes is same and the number of edges varies to a percentage by which we want the two networks to be different. The degree of non-similarity cold is achieved by adding a percentage of edges in the new network considering that the graph still forms a social network and keeps its characteristics [20] [21]. Suppose, if in the base network has N number of nodes and M number of edges, the new network would have N number of nodes and M+M(X/100) number of edges, where X is the degree of non-similarity between the two networks.

There are four scenarios defined in this work to analyze the role of the prior knowledge in community detection in dynamic social networks. This could be achieved by analyzing its effects on some population's adaptation in a network similar to the base network.

- Approach #1: Knowledge and population migration
- Approach #2: Only knowledge migration
- Approach #3: Population migration
- Approach #4: Best individual migration

4.4.1 APPROACH #1: KNOWLEDGE AND POPULATION MIGRATION

This approach is supposed to analyze the ease of adaptation in finding communities by a population that has prior knowledge about an environment with some level of non-similarity. In this approach, both the trained population and its knowledge are imported by the algorithm. This imported population is treated as an initial population for the new environment, and the imported knowledge is assigned as the initial knowledge of the network. Figure 4.6 depicts the algorithm for this approach.

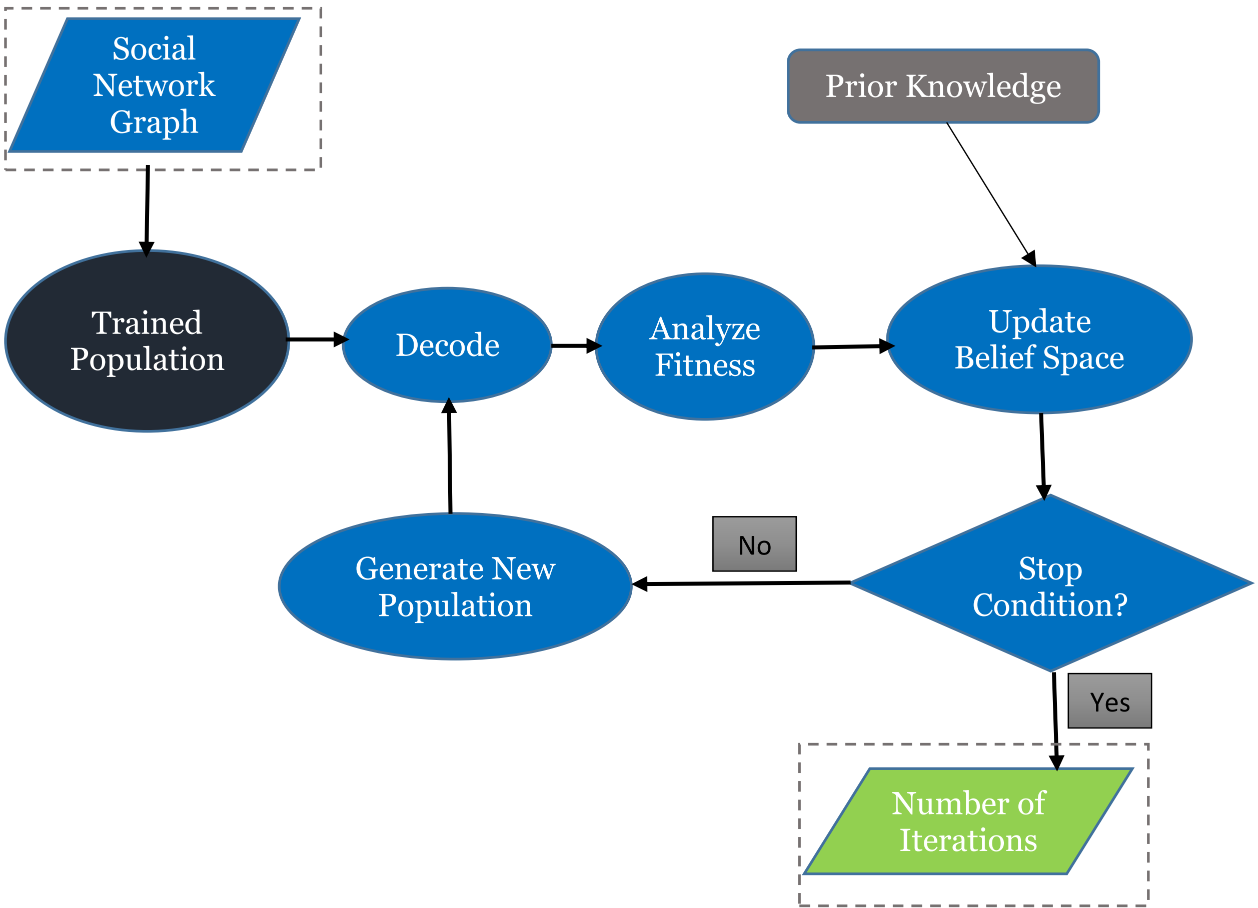

Figure 4.6: Approach #1: Knowledge and Population Migration

4.4.2 APPROACH #2: ONLY KNOWLEDGE MIGRATION

In this approach, only the trained knowledge is imported from the base network to the new non-similar network where the optimization process needs to take place. Here, the initial knowledge of this environment is the migrated knowledge from the base network. But, the initial population of random individuals is generated by the algorithm with the help of imported knowledge. In this approach, the goal is to analyze the impact of only prior knowledge in community detection. Figure 4.7 shows the flowchart of this approach.

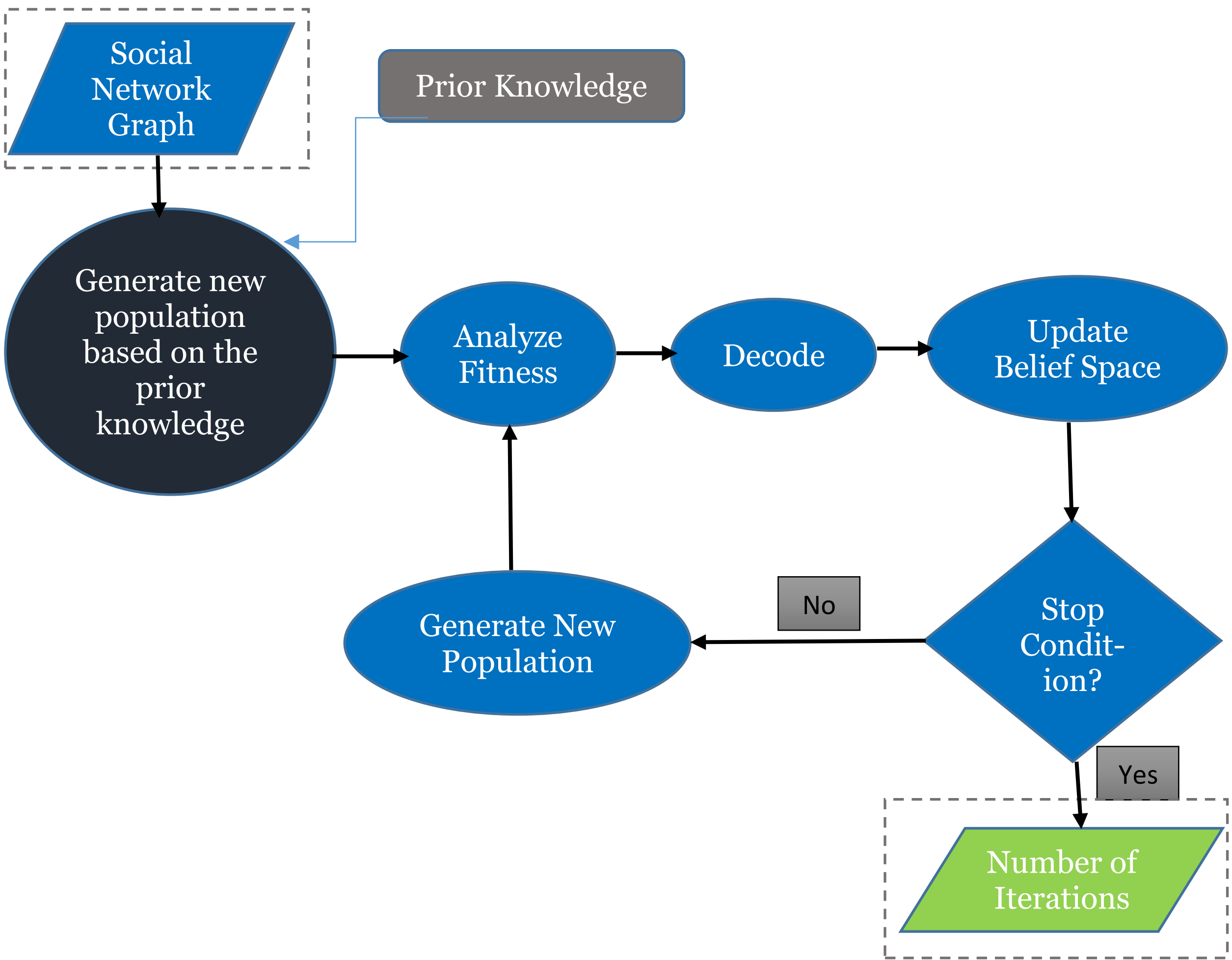

Figure 4.7: Approach #2: Only Knowledge Migration

4.4.3 APPROACH #3: ONLY POPULATION MIGRATION

In this approach, there is an assumption that knowledge is an integral part of any population and it transfers by default when the population migrates to a different environment. Therefore, the imported population is set as the initial population and no random individuals are generated in the initial step of the algorithm. Figure 4.8 shows the pictorial representation of this approach.

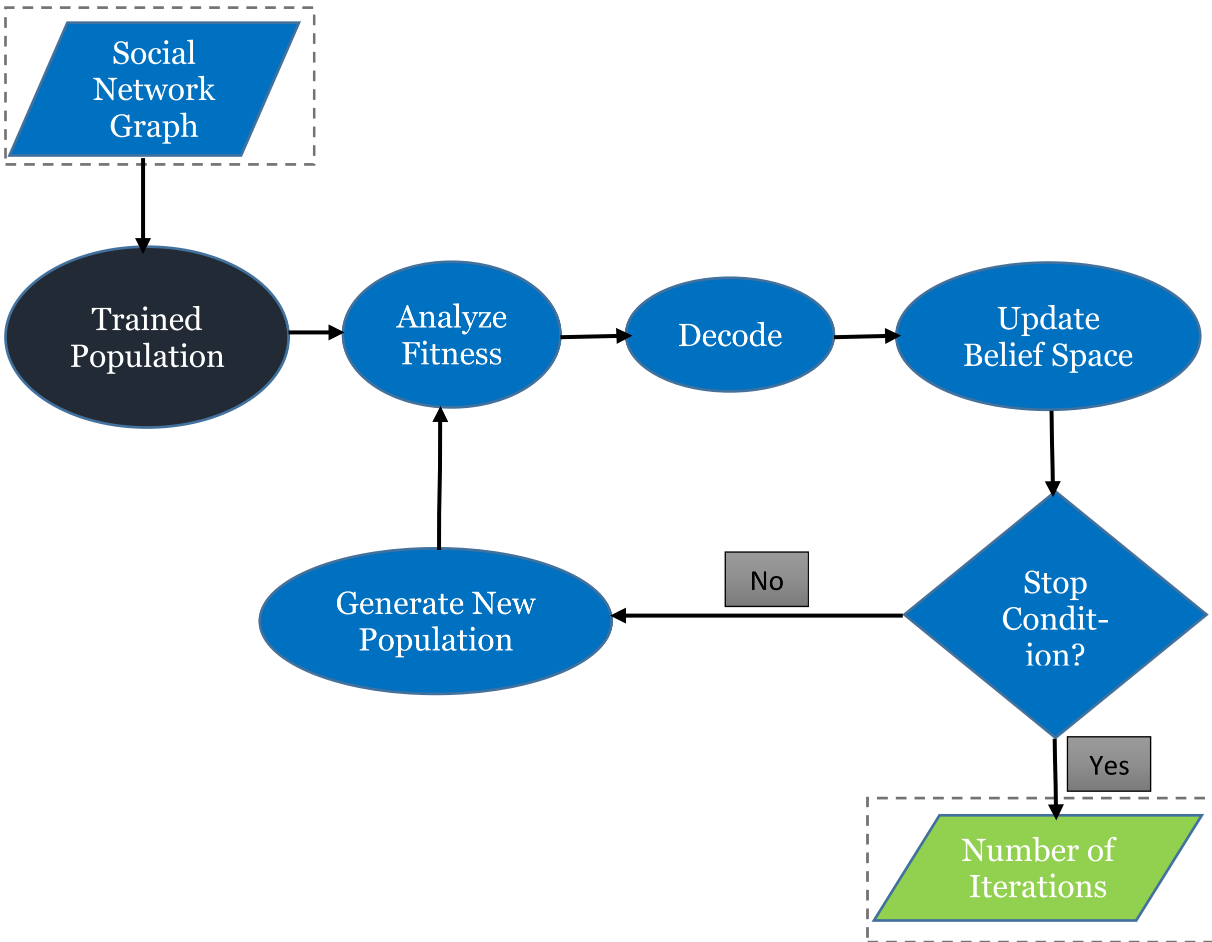

Figure 4.8: Approach #3: Only Population Migration

4.4.4 APPROACH #4: BEST INDIVIDUAL MIGRATION

In this approach, again, we assume that knowledge is an integral part of individuals and it migrates by default when these individuals migrate from one network to another. This scenario is based on the migration of the best individuals from the base network to the other network with some level of non-similarity. The initial population of the new network is created using a combination of the imported best individuals and some randomly generated individuals by the algorithm. Figure 4.9 depict the flowchart of this approach.

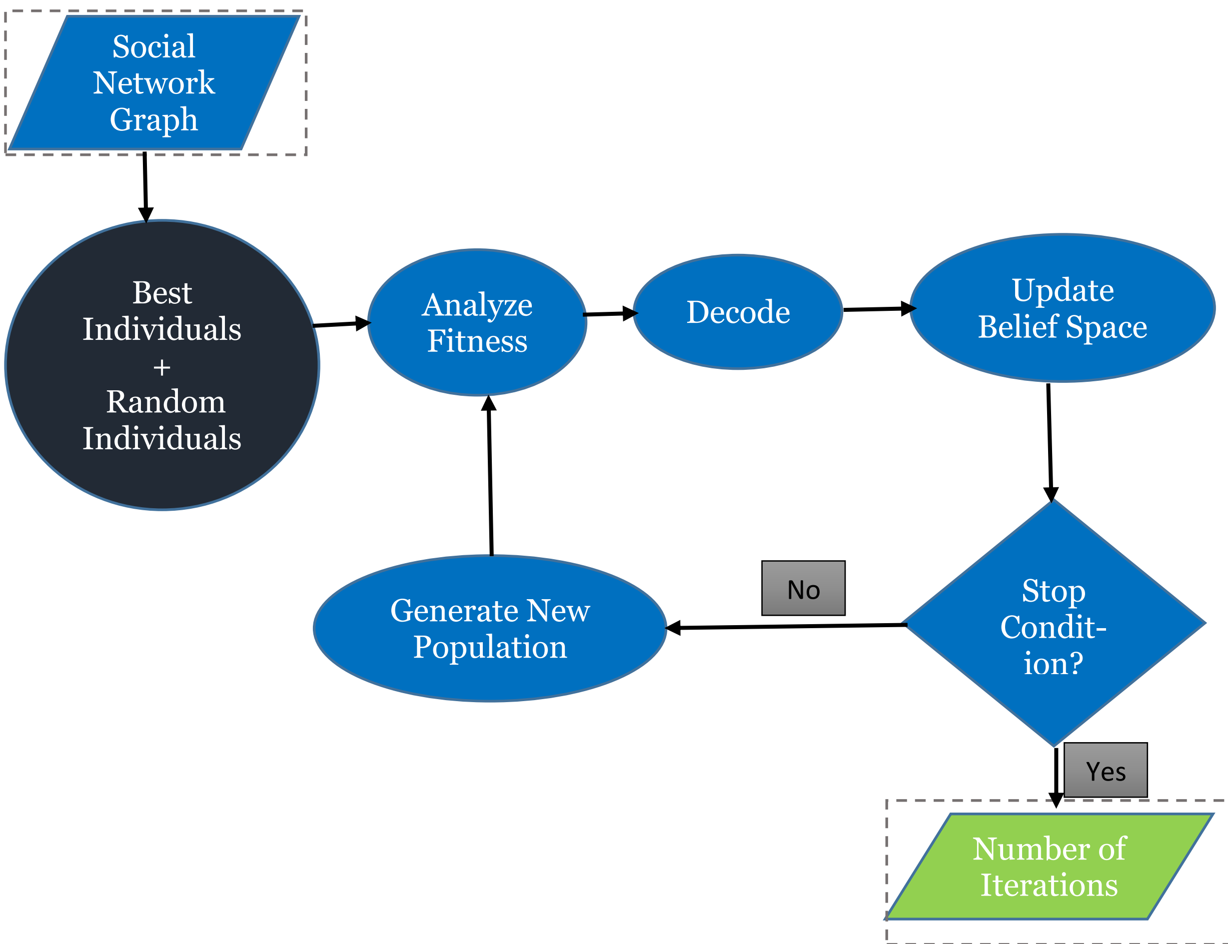

Figure 4.9: Approach #4: Best Individual Migration

CHAPTER 5
EXPERIMENTAL RESULTS

This chapter discusses the experimental results for detecting the role of prior knowledge in MPCA for community detection in dynamic social networks. The first part of this chapter explains the implementation details used to test our research. It also explains the testing environment and the benchmark graphs used in this work. The next chapter is about the experiments and details related to them. The last part of this chapter does a descriptive analysis and discussion about the results.

5.1 IMPLEMENTATION DETAILS

To test the different scenarios explained before and analyze the role of prior knowledge, MPCA proposed in [10] has been implemented in this work. Here, the size of each population has been set to 200 for our experiments. For this work, we have used community score as fitness function defined by Pizzuti in [5] and explained before in the previous chapter. The selection rate for this process is set to 20%. The maximum number of iteration for detecting communities is set to 50.

One of the most widely accepted benchmark networks proposed by Girvan et al. in [1] has been used to generate ten social network graphs for our base networks in all four approaches. In each base network, there are 128 nodes such that each network is divided into four communities with 32 nodes in each. The average degree of each node is set to 16 such a way that the sum of all the edges of a node inside a community (Z_{in}) and outside it (Z_{out}) is equal to 16 i.e. Z_{in} + Z_{out} =

16. According to [1], more the value of Z_{out}, more will be its complexity. So, here we have set the average value of Z_{out} to be 5.

Furthermore, for this work the non-similar networks are defined in five different levels of non-similarity- 5%, 10%, 15%, 20%, and 25% of the base network. As discussed before, this means that the base network and the new network will have the same number of nodes, but the number of edges will vary with one of the percentages of non-similarity. Say the base network has 100 nodes and 200 edges and the degree of non-similarity is 5%, then the new network will have 100 nodes and 210 edges in it. These levels of non-similarity among the networks are used for the four approaches we have defined.

To find the best results, for all the ten base networks ten more networks each of non-similarity 5%, 10%, 15%, 20%, 25% are created. So, in total 500 networks are designed under each approach to analyzing the outcome.

5.2 EXPERIMENTAL RESULTS

We conducted experiments on the four scenarios to understand the role of prior knowledge. All the experimental results are described in the next subsections.

5.2.1 Approach #1

In this approach, the community detection algorithm was run on the base network. Then the normative knowledge matrix and the trained population of this base network were imported by the algorithm and used as the new network's initial knowledge and population respectively. Algorithm explained in approach #1, and the normal community detection algorithm was run ten times on all the 50 networks. Table 5.1 shows the average number of iterations to detect the communities in networks having no prior knowledge of the base network and the average number of iterations to detect the communities when the prior knowledge was applied to these networks.

Figure 5.1 shows the graphical implementation of the results found in the table below.

Base Network (BN)	Network with no Prior Knowledge	Network 5% non-similar to BN	Network 10% non-similar to BN	Network 15% non-similar to BN	Network 20% non-similar to BN	Network 25% non-similar to BN
	Average Number of Iterations					
BN #1	26	1	6	7	23	46
BN #2	25	1	6	7	19	20
BN #3	26	1	11	16	16	25
BN #4	23	1	5	8	16	25
BN #5	27	1	9	18	16	21
BN #6	21	1	15	10	16	21
BN #7	20	1	4	18	19	28
BN #8	23	1	27	3	32	28
BN #9	23	1	7	20	35	35
BN #10	22	1	6	13	15	24

Table 5.1: Average number of iterations for community detection when both population and knowledge are migrated (Approach #1) from base network to other non-similar networks

5.2.2 Approach #2

Similar to the first approach, the community detection algorithm was run on the base networks, and the normative knowledge matrix was extracted. Then we imported this knowledge on the algorithm defined in approach #2. After this, we ran both the algorithms (the one with prior knowledge defined in approach #2, and the normal community detection algorithm) 10 times each on all the 50 networks. As shown in Table 5.2, some iterations to find the correct communities in the case where no prior knowledge has been shared is almost fixed. But when the knowledge is shared among the non-similar networks, iterations have a significant positive impact in most of the scenarios. Figure 5.2 shows the graph view of the corresponding results.

Base Network (BN)	Network with no Prior Knowledge	Network 5% non-similar to BN	Network 10% non-similar to BN	Network 15% non-similar to BN	Network 20% non-similar to BN	Network 25% non-similar to BN
	Average Number of Iterations					
BN #1	23	1	8	8	25	19
BN #2	27	1	12	18	20	25
BN #3	23	1	9	18	20	29
BN #4	23	1	14	17	20	37
BN #5	25	1	16	22	16	42
BN #6	25	1	13	16	18	19
BN #7	27	1	11	16	19	26
BN #8	31	1	18	15	22	26
BN #9	24	1	7	11	18	25
BN #10	23	1	15	16	28	29

Table 5.2: Average number of iterations for community detection when only knowledge is migrated (Approach #2) from the base network to other non-similar networks

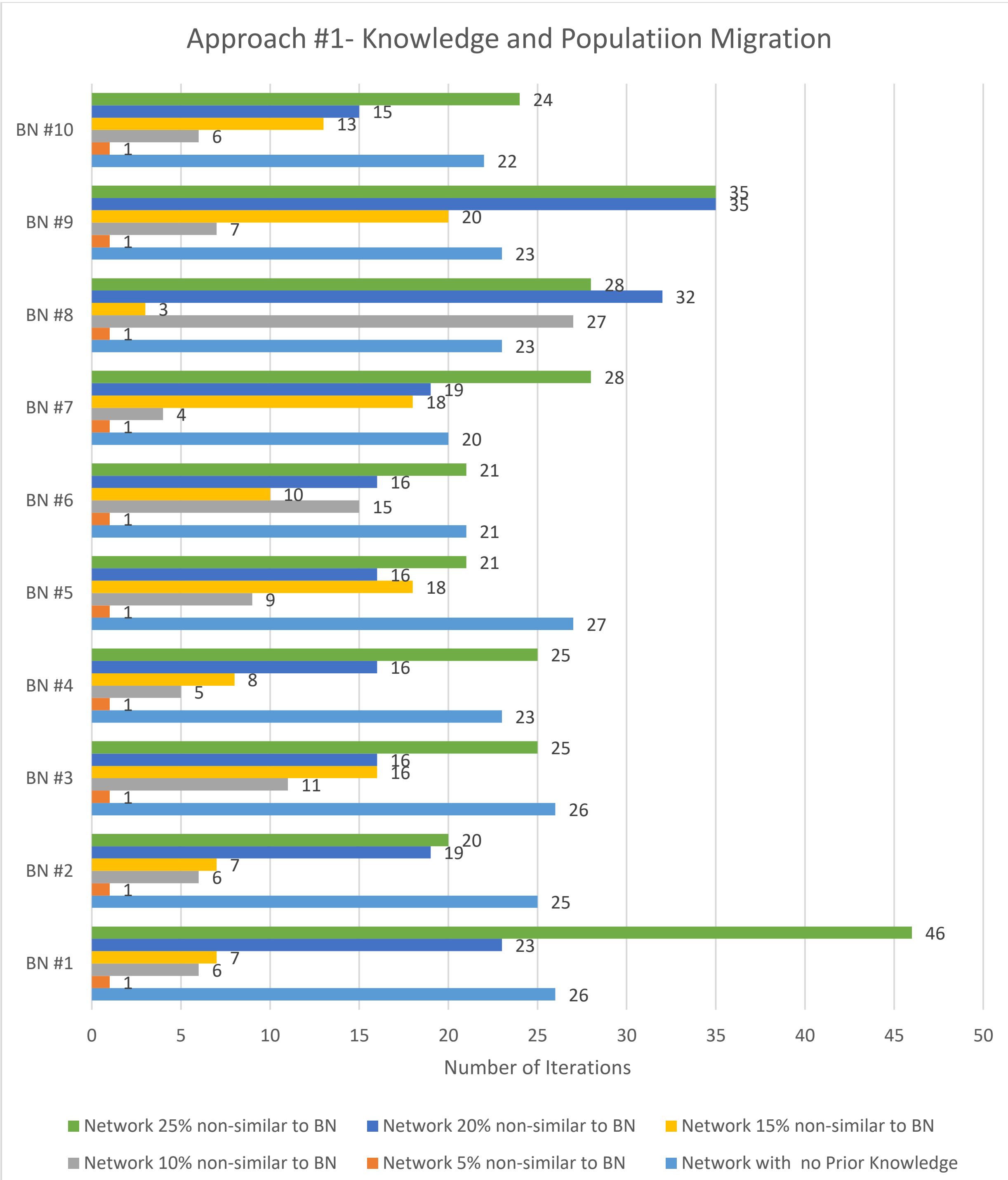

Figure 5.1: Approach #1- Knowledge and Population Migration

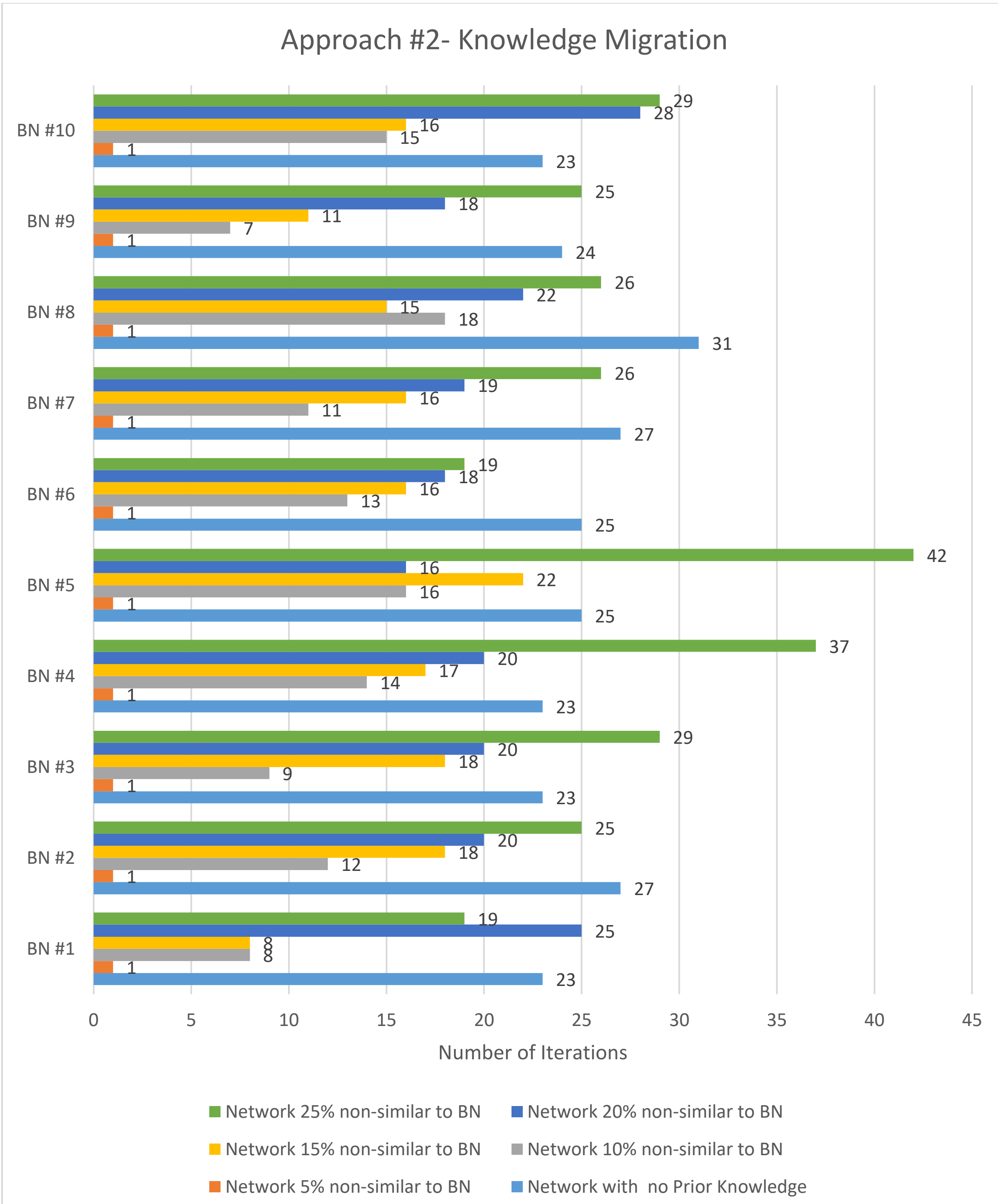

Figure 5.2: Approach #2- Knowledge Migration

5.2.3 Approach #3

In this scenario, to identify the role of prior knowledge, the first step is same as what we did in the previous approaches. We run the community detection algorithm on the base network, and the best population is imported by the algorithm described in approach #3 as its initial population. Two community detection algorithms are run separately- one with the previous population and without it ten times each on all the 50 networks. Table 5.3 shows the number of iterations to detect communities when no previous population was applied and when it was applied to the non-similar networks. Furthermore, Figure 5.3 shows the graphical representation of the results we achieved in this scenario.

5.2.4 Approach #4

In the fourth approach, after the initial process of community detection in the base networks, only the best individuals are transferred to the new non-similar networks according to the algorithm defined in the previous chapter. We transfer only 40% of the required individuals from the base network which have been defined as the best among that population. Rest of the individuals are randomly generated from the graph using the process described under the chapter design and methodology. Table 5.4 showed the number of iterations when no best individuals were applied in the initial population and when they were applied to find the communities in the network. And, Figure 5.4 depicts the graphical view of the results.

Base Network (BN)	Network with no Prior Knowledge	Network 5% non-similar to BN	Network 10% non-similar to BN	Network 15% non-similar to BN	Network 20% non-similar to BN	Network 25% non-similar to BN
	Average Number of Iterations					
BN #1	23	1	5	10	18	25
BN #2	23	1	6	11	18	22
BN #3	25	1	6	11	22	23
BN #4	23	1	7	9	21	25
BN #5	26	1	7	14	15	26
BN #6	23	1	6	14	19	27
BN #7	25	1	9	12	24	27
BN #8	27	1	11	9	17	26
BN #9	23	1	7	26	19	29
BN #10	26	1	10	10	22	29

Table 5.3: Average number of iterations for community detection when the only population is migrated (Approach #3) from the base network to other non-similar networks

Base Network (BN)	Network with no Prior Knowledge	Network 5% non-similar to BN	Network 10% non-similar to BN	Network 15% non-similar to BN	Network 20% non-similar to BN	Network 25% non-similar to BN
	Average Number of Iterations					
BN #1	25	1	8	9	19	22
BN #2	27	1	6	8	22	29
BN #3	27	1	5	14	21	22
BN #4	28	1	5	9	17	24
BN #5	27	1	12	9	19	25
BN #6	26	1	5	14	17	25
BN #7	24	1	9	12	18	25
BN #8	25	1	8	13	19	26
BN #9	25	1	8	9	23	27
BN #10	23	1	9	8	23	22

Table 5.4: Average number of iterations for community detection when only best individuals are migrated (Approach #4) from the base network to other non-similar networks

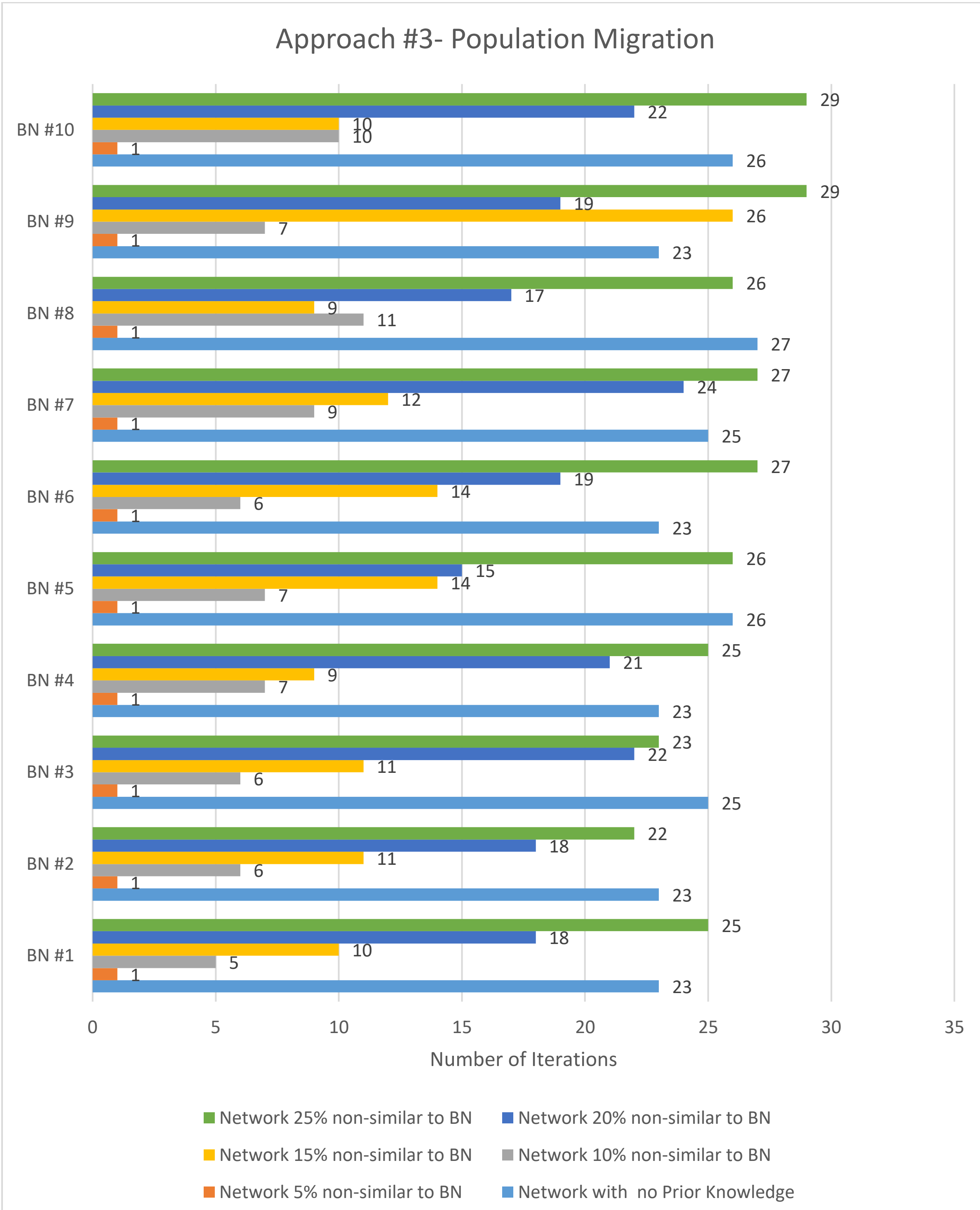

Figure 5.3: Approach #3- Population Migration

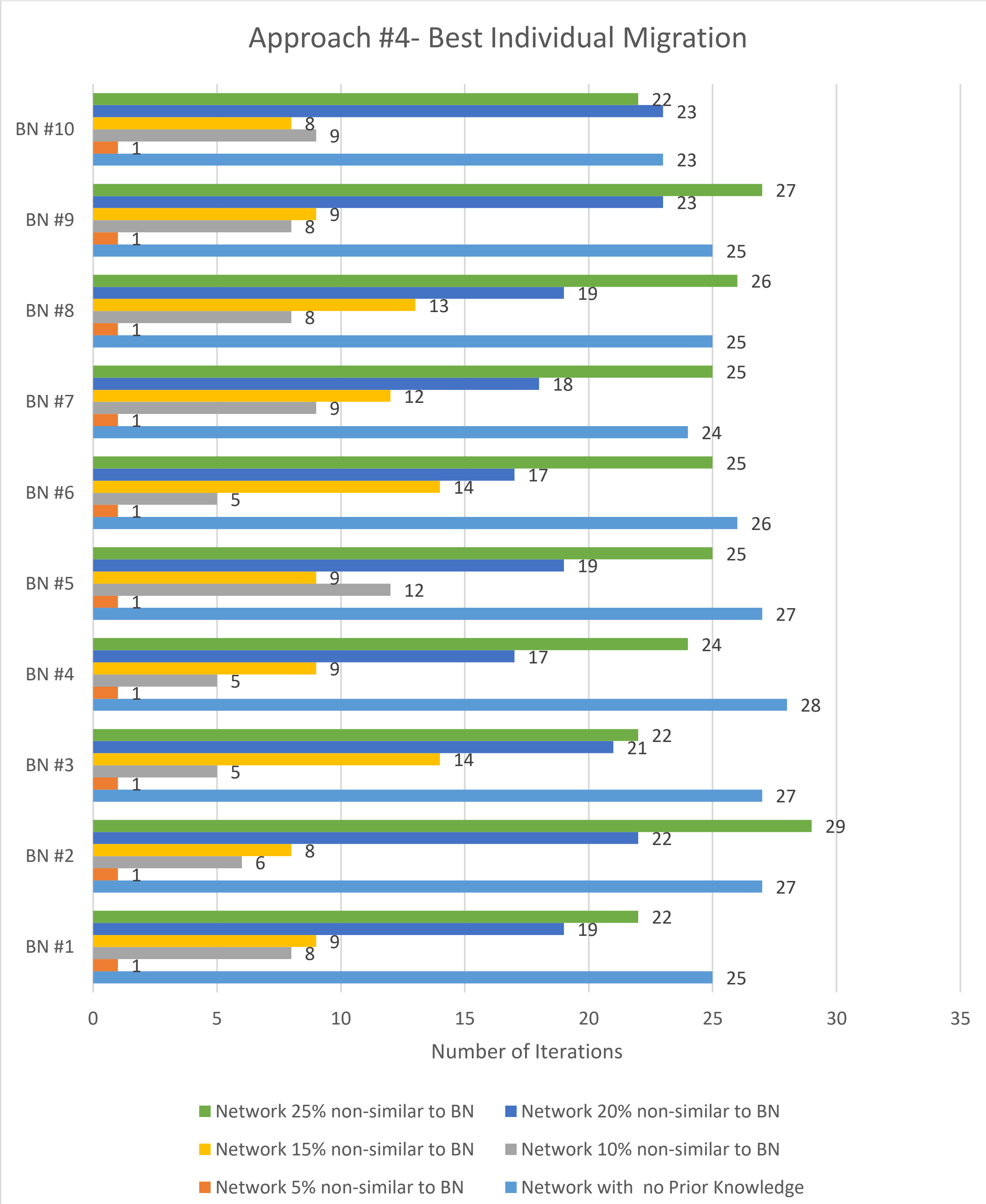

Figure 5.4: Approach #4- Best Individual Migration

5.3 ANALYSIS AND DISCUSSION

The main aim of our research was to analyze the role of prior knowledge in MPCA for community detection process in dynamic social networks. Here we challenged ourselves to check the feasibility of prior knowledge and how it can be accessed. We briefly discussed how some prior knowledge from a network could affect the adaptation process of a new population in a similar environment. How prior knowledge about some environment can help a population adapt to the new environment.

5.3.1 COMPARISON OF SCENARIOS

Besides the fact that all the approaches defined here are different from each other, there is some similarity between them. Such as, all these approaches are based on the concept of prior knowledge and deal with the community detection problem in some complex networks. We calculate the effectiveness of these heuristics with the help of a number of iteration involved in the community finding process. As we deeply analyze the tables 5.1 to 5.4, we can say they are comparable.

Here, we present the comparison between all the four scenarios in Table 5.5 and Figure 5.5.

	A1	A2	A3	A4
No prior knowledge attached	23.6	25.1	24.4	25.7
5% Non-similarity with Base Network	1	1	1	1
10% Non-similarity with Base Network	9.6	12.3	7.4	7.5
15% Non-similarity with Base Network	12	15.7	12.6	10.5
20% Non-similarity with Base Network	20.7	20.6	19.5	19.8
25% Non-similarity with Base Network	27.3	27.7	25.9	24.7

Table 5.5: Comparative result showing average number of iterations in all four approaches

Based on the results obtained in Figure 5.5 it is evident that all the different methods proposed for knowledge transfer are not same. Some have less and some have comparatively more time complexity. Comparing all the approaches under the scenario where the non-similarity between the networks is 5%, we see that an average number of iterations for community detection in all the four approaches is 1. The reason behind this is both the networks are only 5% non-similar, or we can say 95% similar. In this case knowledge migration and population, adaptation looks very feasible. It means prior knowledge can be given by any of the four methods proposed in this work will be sufficiently enough for community detection in a linear time.

On the other hand, when the degree of non-similarity between the training network and the target networks is 10% we can see that the best population transfer (approach #3) and the best individual transfer (approach #4) are better than both knowledge and population migration (approach #1) from the base network to the non-similar networks. Also, these three approaches are much better than approach #2 i.e. best knowledge migration.

Now, if we analyze the next approach, when the degree of non-similarity between the two networks increases to 15%, we can clearly say that the transfer of only best individuals (approach #4) from the base network to the non-similar network is quite better (value is 10.5) in finding communities as compared to approach #1 where we transfer both, the knowledge and the population between the training network and the target network. Furthermore, approach #4 and #1 are better than the approach #3 where we transfer only best population between the networks. This could be because a set of best individuals will always have a better fitness value than a whole best population (where some individuals might not be that fit). At last, all these approaches show better results than the second approach where only the knowledge is imported from the base network to its non-similar network (approach #2).

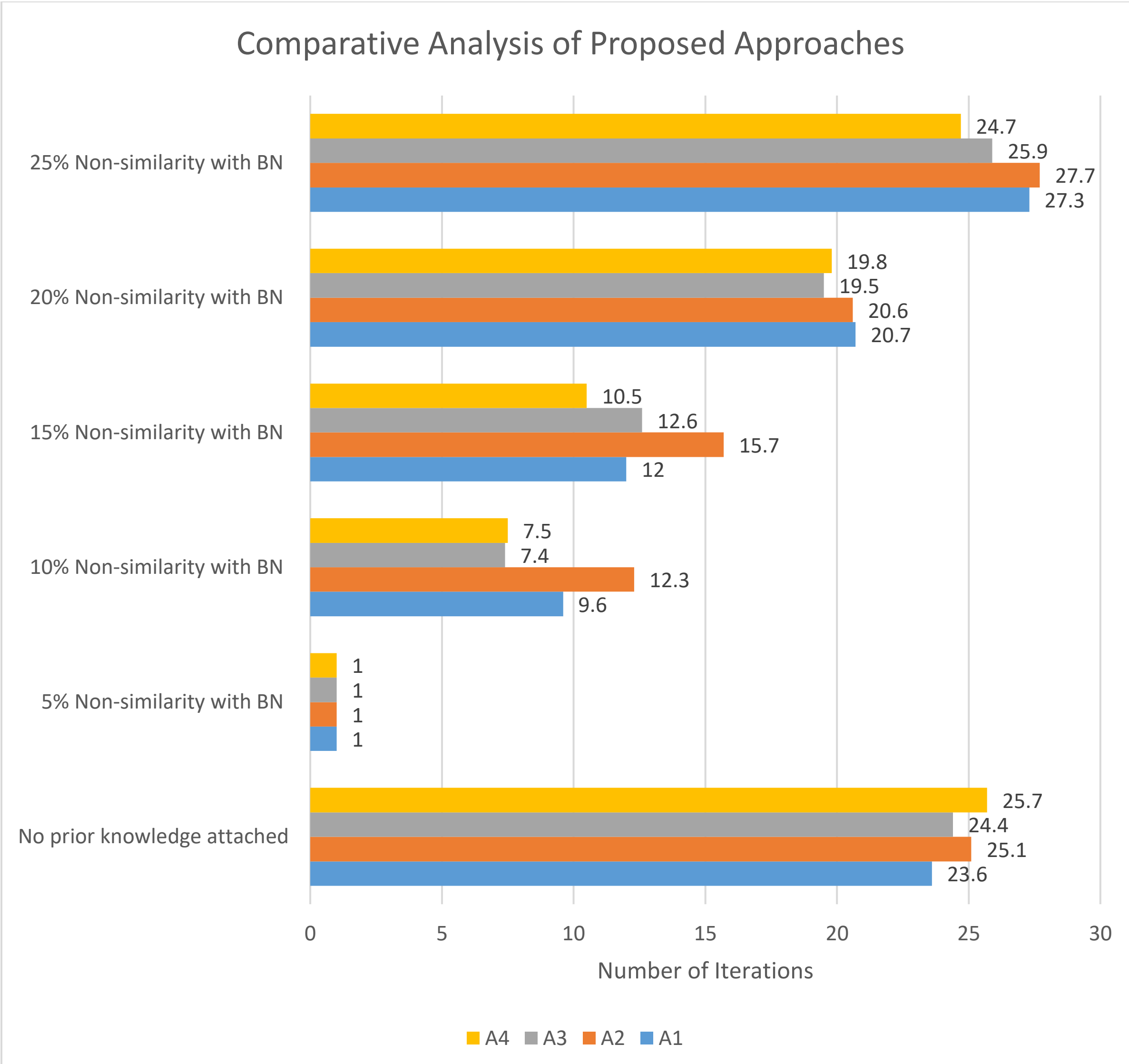

Figure 5.5: Comparative analysis of all the proposed approaches

Analyzing the next approach, where the degree of non-similarity between the training network and the target network increases to 20%, we can see that there is not much difference between all the proposed approaches. The average number of iterations for the approaches lie in the range of 19.5 to 20.7 with a very little different between any two approaches. Still, we can see that the performance of approach #3 is better than approach #2 where only the knowledge is transferred from the base network to the non-similar network.

25% non-similarity between the base network and their corresponding target network does not look promising enough to solve the adaptation problem with fewer efforts than solving it without any prior knowledge. In fact, in most of the approaches it is apparent with the data values 23.6, 25.1, 24.4, 25.7 for non-similar network without any prior knowledge and 27.3, 27.7, 25.9 and 24.7 for the target network with 25% non-similarity and prior knowledge, that the number of iterations to solve the adaptation problem with 25% or more non-similarity is costlier than not using any prior knowledge at all.

Finally, we can say that prior knowledge is helpful to population adoption process if the degree of non-similarity between the networks is less than 25%. Analyzing our proposed four approaches, it is evident that only prior knowledge is not helpful for problem optimization every time. In fact, we have analyzed that most of the time a trained population or a group of fit individuals could be more helpful in population adaptation process than just a prior knowledge.

5.3.2 TREND

Plotting a graph (Figure 5.6) between the number of iterations to solve the adaptation problem and the four different levels of non-similarities described in this work, we can see that the trend shows a linear pattern. But, there is a catch. If we increase the order of non-similarity between the networks, it is hard to solve the problem of community detection using the prior knowledge of the training network because the two networks are so different that the prior knowledge does not help to converge the solution space. So, it is not a linear trend, but an exponential one.

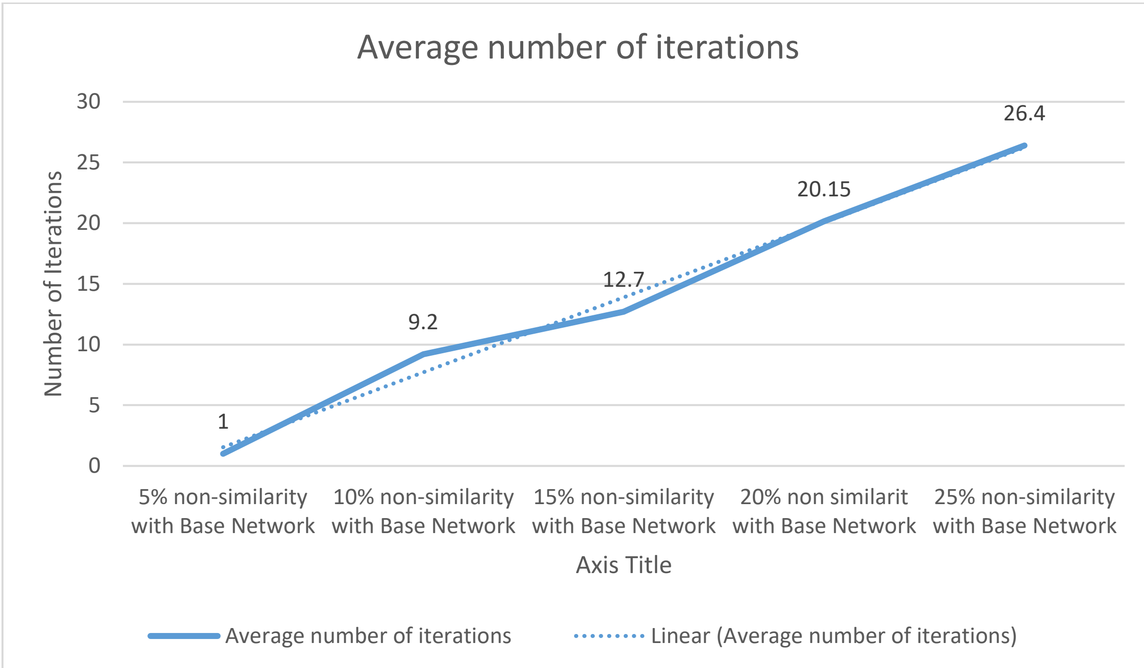

Figure 5.6: Trend

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 CONCLUSION

Since the inception of the internet, it has gained an exponential growth in the number of users and its availability. According to some data projections, as per the year 2016, there were more than 2.3 billion social network users worldwide, and it is supposed to grow up to 2.67 billion users by the year 2018 [29]. This huge growth in the social networks has created a need to fast and efficient networks. Communities are an essential part of any social network. If we need to understand the social networks, we need to understand the communities and their structure in more detail. Detecting efficient communities using efficient algorithms could help us understanding the social networks much better [1] [6] [17].

As explained in multi-population cultural algorithms, knowledge is one essential part of any community [17]. It differentiates a normal network from the other in a sense that a population having knowledge can evolve more quickly as compared to the one which does not possess it. Knowledge helps to converge the solutions much fast and more accurately.

MPCA is one of the best heuristics to solve the problem of community detection in dynamic social networks. In this work, we have proposed four different approaches to import knowledge from one network to another network (where, these two networks are similar to some extent), so we can analyze the role of this knowledge (called prior knowledge) in the process of community detection. The four different approaches defined in this work are-

- Approach #1- where we transfer both the knowledge and population from the base network to the target network.
- Approach #2- where we transfer the knowledge from the base network to the target network.
- Approach #3- which explains about the transfer of best population from the training network to the target network assuming every individual has some knowledge about its network.
- In Approach #4 we transfer only some best individuals from the base population to the target population having the same assumption that every individual has some knowledge about its network by default.

Furthermore, we also defined the networks that are similar to some extent and called them non-similar networks. The degree of non-similarity (between 5% and 25%) defines how much the training and the target networks are similar to each other.

After analyzing the results, we can say that prior knowledge helps to find to communities in networks that are similar to some degree with the maximum level of non-similarity as 25%. If the two networks are more similar than 75% to each other, there are high chances the prior knowledge of the base networks would help the target network to find the communities in it in fewer iterations, but if the degree of non-similarity is more than or equal to 25% it would not only do minimal help in finding the communities it would actually slow the process down to find communities, and it will be very hard for the training network to converge the solution space for the target network. Furthermore, we have analyzed one interesting fact that, instead of using only knowledge as prior knowledge, trained populations and individuals are much better in finding communities among the networks.

6.2 FUTURE WORK

The future work of this proposed analysis should be an extension of the real-world networks. This proposed work is implemented in the computer-generated environment. Furthermore, in the future, we can implement this work on some more complex networks involving some more nodes and links which could increase the complexity many folds. We would also like to add some more test cases using different kinds of knowledge for the belief space.

REFERENCES

[1] Girvan, Michelle, and Mark EJ Newman. "Community structure in social and biological networks." *Proceedings of the national academy of sciences* 99, no. 12 (2002): 7821-7826.

[2] Lancichinetti, Andrea, and Santo Fortunato. "Community detection algorithms: a comparative analysis." *Physical review E* 80, no. 5 (2009): 056117.

[3] Reynolds, Robert G. "An introduction to cultural algorithms." In *Proceedings of the third annual conference on evolutionary programming*, vol. 131139. Singapore, 1994.

[4] Lancichinetti, Andrea, Filippo Radicchi, José J. Ramasco, and Santo Fortunato. "Finding statistically significant communities in networks." *PloS one* 6, no. 4 (2011): e18961.

[5] Pizzuti, Clara. "GA-Net: A genetic algorithm for community detection in social networks." *Parallel problem solving from nature–PPSN X* (2008): 1081-1090.

[6] Newman, Mark EJ. "The structure and function of complex networks." *SIAM review* 45, no. 2 (2003): 167-256.

[7] Newman, Mark EJ, and Michelle Girvan. "Finding and evaluating community structure in networks." *Physical review E* 69, no. 2 (2004): 026113.

[8] Lin, Yu-Ru, Yun Chi, Shenghuo Zhu, Hari Sundaram, and Belle L. Tseng. "Facetnet: a framework for analyzing communities and their evolutions in dynamic networks." In *Proceedings of the 17th international conference on World Wide Web*, pp. 685-694. ACM, 2008.

[9] De Meo, Pasquale, Emilio Ferrara, Giacomo Fiumara, and Alessandro Provetti. "Generalized louvain method for community detection in large networks." In *Intelligent Systems Design and Applications (ISDA), 2011 11th International Conference on*, pp. 88-93. IEEE, 2011.

[10] Zadeh, Pooya Moradian, and Ziad Kobti. "Community detection in social networks by cultural algorithm." In *Collaboration Technologies and Systems (CTS), 2015 International Conference on*, pp. 319-325. IEEE, 2015.

[11] Guo, Yi-nan, Jian Cheng, Yuan-yuan Cao, and Yong Lin. "A novel multi-population cultural algorithm adopting knowledge migration." *Soft computing* 15, no. 5 (2011): 897-905.

[12] Hlynka, Andrew William, and Ziad Kobti. "Knowledge Sharing Through Agent Migration with Multi-Population Cultural Algorithm." In *FLAIRS Conference*. 2013.

[13] Saleem, Saleh, and Robert Reynolds. "Cultural algorithms in dynamic environments." In *Evolutionary Computation, 2000. Proceedings of the 2000 Congress on*, vol. 2, pp. 1513-1520. IEEE, 2000.

[14] Kobti, Ziad. "A Knowledge-Migration-Based Multi-Population Cultural Algorithm to Solve Job Shop Scheduling." In *Twenty-Fifth International FLAIRS Conference*. 2012.

[15] Mokom, Felicitas, and Ziad Kobti. "Improving artifact selection via agent migration in multi-population cultural algorithms." In *Swarm Intelligence (SIS), 2014 IEEE Symposium on*, pp. 1-8. IEEE, 2014.

[16] He, Xinyu. "Heuristics for Multi-Population Cultural Algorithm." (2015).

[17] Peng, Bin, and Robert G. Reynolds. "Cultural algorithms: Knowledge learning in dynamic environments." In *Evolutionary Computation, 2004. CEC2004. Congress on*, vol. 2, pp. 1751-1758. IEEE, 2004.

[18] Chung, Chan-Jin, and Robert G. Reynolds. "CAEP: An evolution-based tool for real-valued function optimization using cultural algorithms." *International Journal on Artificial Intelligence Tools* 7, no. 03 (1998): 239-291.

[19] Reynolds, Robert G., and S. M. Saleem. "The impact of environmental dynamics on cultural emergence." *Perspectives on Adaptions in Natural and Artificial Systems* (2005): 253-280.

[20] Zadeh, Pooya Moradian, Mukund Pandey, and Ziad Kobti. "A study on population adaptation in social networks based on knowledge migration in cultural algorithm." In *Evolutionary Computation (CEC), 2016 IEEE Congress on*, pp. 4405-4412. IEEE, 2016.

[21] Watts, Duncan J., and Steven H. Strogatz. "Collective dynamics of 'small-world' networks." *nature* 393, no. 6684 (1998): 440-442.

[22] Blondel, Vincent D., Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. "Fast unfolding of communities in large networks." *Journal of statistical mechanics: theory and experiment* 2008, no. 10 (2008): P10008.

[23] Jin, Xidong, and Robert G. Reynolds. "Using knowledge-based evolutionary computation to solve nonlinear constraint optimization problems: a cultural algorithm approach." In *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on*, vol. 3, pp. 1672-1678. IEEE, 1999.

[24] Lancichinetti, Andrea, Santo Fortunato, and János Kertész. "Detecting the overlapping and hierarchical community structure in complex networks." *New Journal of Physics* 11, no. 3 (2009): 033015.

[25] Lancichinetti, Andrea, Santo Fortunato, and Filippo Radicchi. "Benchmark graphs for testing community detection algorithms." *Physical review E* 78, no. 4 (2008): 046110.

[26] Eiben, Agoston E., and James E. Smith. *Introduction to evolutionary computing*. Vol. 53. Heidelberg: springer, 2003.

[27] Yang, Jaewon, Julian McAuley, and Jure Leskovec. "Community detection in networks with node attributes." In *Data Mining (ICDM), 2013 IEEE 13th international conference on*, pp. 1151-1156. IEEE, 2013.

[28] Hlynka, Andrew William. "Using Heritage in Multi-Population Evolutionary Algorithms." Ph.D. diss., University of Windsor (Canada), 2016.

[29] URL: https://www.statista.com/statistics/278414/number-of-worldwide-social-network-users/

[30] Lancichinetti, Andrea, Santo Fortunato, and Filippo Radicchi. "Benchmark graphs for testing community detection algorithms." *Physical review E* 78, no. 4 (2008): 046110.

[31] Handl, Julia, and Joshua Knowles. "An evolutionary approach to multiobjective clustering." *IEEE transactions on Evolutionary Computation* 11, no. 1 (2007): 56-76.

[32] Kobti, Ziad. "A Knowledge-Migration-Based Multi-Population Cultural Algorithm to Solve Job Shop Scheduling." In *Twenty-Fifth International FLAIRS Conference*. 2012.

VITA AUCTORIS

Mukund Pandey was born in 1988 in Pune, India. He obtained a Bachelor degree in Computer Science from Uttar Pradesh Technical University, Lucknow, India in 2011. He is currently a candidate for the Master's degree in Computer Science at the University of Windsor and hopes to graduate in Spring 2017.