9-26-2018

Improving Quality of the Solution for the Team Formation Problem in Social Networks Using SCAN Variant and Evolutionary Computation

Amangel Bhullar
University of Windsor

Improving Quality of the Solution for the Team Formation Problem in Social Networks Using SCAN Variant and Evolutionary Computation

by

Amangel Bhullar

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
2018

Improving Quality of the Solution for the Team Formation Problem in Social Networks Using SCAN Variant and Evolutionary Computation

by

Amangel Bhullar

APPROVED BY:

R. J. Urbanic,

Department of Mechanical, Automotive and Materials Engineering

---

M. Kargar,

School of Computer Science

Z. Kobti, Advisor

School of Computer Science

P. Moradian Zadeh, Co-Advisor

School of Computer Science

September 12, 2018

Declaration of Co-Authorship/Previous Publication

1. Co-Authorship

I hereby declare that this thesis incorporates material that is the result of research conducted under the supervision of Dr. Ziad Kobti (Advisor) and Dr. Pooya Moradia Zadeh (Co-Advisor). In all cases, the key ideas, primary contributions, experimental designs, data analysis and interpretation, were performed by the author, and the contribution of co-authors was primarily through the proofreading of the published manuscripts. Dr. Mehdi Kargar and Kalyani Selvarajah contributed in collecting data and explaining the materials.

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 product of my own work.

2. Previous Publication

This thesis includes two original papers that has been previously submitted in peer reviewed conferences, as follows:

Section	Full Citation	Publication status
3.3, 3.3.1 and 3.3.3	Kalyani Selvarajah, Amangel Bhullar, Dr. Ziad Kobti and Dr. Mehdi Kargar ."Weighted-SCAN Clustering Algorithm For Finding; a Team of Experts in Social Networks". 31st International FLAIRS (The Florida Association for Artificial Intelligence Research Society) Conference / Association of Advanced Artificial Intelligence (AAAI) 2018 (pp. 209-212).	Accepted

Section	Full Citation	Publication status
3.6, 3.6.1 and 3.6.2	Amangel Bhullar, Kalyani Selvarajah, Dr. Ziad Kobti and Dr. Mehdi Kargar. "Hybrid Genetic Algorithm based Approach For Finding; the Team of Experts in a Social Networks". 17th IEEE International Conference On Machine Learning And Applications / IEEE ICMLA 2018 .	Submitted

3. General

I certify 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.

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

Social Network Analysis helps to visualize and understand the roles and relationships that ease or impede the collaboration and sharing of the information and knowledge in an organization. In this research work, we will focus on the Team Formation Problem (TFP) which is an open problem where we need to identify an ideal team, with members of complementary talent or skills, to solve any given task. Current research suggests that TFP solutions have been attempted with evolutionary computation approach using Cultural Algorithms (CA) and Genetic Algorithms (GA). However, SCAN (Structural Clustering Algorithm for Networks) variants such as WSCAN (Weighted Structural Clustering Algorithm for Networks) demonstrate a high capability to find solutions for another type of network problems. In this thesis, we first propose to use WSCAN-TFP algorithm to deal with the problem of team formation in social networks, and we our findings indicate that WSCAN-TFP algorithm worked faster than the evolutionary algorithms counterparts but was of lower performance compared to CAs and GAs. Next, we propose two hybrid solutions by combining GA and CA with a modified WSCAN-TFP algorithm. To test the performance of our proposed approaches, we define multiple quality criteria based on communication cost (CC), average fitness score (AFS) and average processing time. We used big datasets from DBLP nodes network with sizes 50K and 100K. The results show that our proposed methods HGA and HCA can find the near-optimal solutions faster with minimum communication cost with the improvement of $\approx 66\%$ and $\approx 57\%$ in average fitness in comparison to existing GA and CA methods respectively.

Dedication

I would like to dedicate this thesis to my family.

Father: Babu Singh Bhullar

Mother: Manjit Kaur Bhullar

Sister: Simrandeep Kaur

Acknowledgements

There are many people to whom I would like to acknowledge for their help and support for my journey of the master thesis.

First and foremost I would pay my gratitude to my supervisor Dr. Ziad Kobti. Under his guidance, I had enjoyed a lot working on my research work. It was a great pleasure to work and discuss with him. Without his support, this won't have been possible. I would also like to appreciate the amount of time he invested in me, the funding he provided and also the knowledge he shared with me.

In addition to this, many thanks to my committee members Dr. Pooya Moradian Zadeh, Dr. Jill Urbanic and Dr. Mehdi Kargar for their valuable time and their support. I would like to express my appreciation to Ms. Gloria Mensah, Ms. Karen Bourdeau, and Ms. Margaret Garabon who always supported me when I needed assistance in various academic issues.

I would like to thank my parents for their counsel and the sympathetic ear. They are always there for me and support me in all possible ways. Next, I would like to thank my sister who helped me a lot during my study with her wise counsel and her knowledge.

I am also very thankful to my friends for their moral support and listening to my problems for long hours.

Finally, I would like to thank God for unconditional love.

Amangel Bhullar

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Problem Definition

Social Network Analysis (SNA) has many open problems to solve such as Team Formation Problem (TFP), Link Prediction, Leadership Detection, Community Detection, Migration between communities, Influence Analysis, Sentimental Analysis, Collaborative Recommendation, and Fraud Detection.

TFP involves finding an optimal solution to assemble a team to complete a task (T), which has a set of k required skills denoted by some criteria. The team (X') is selected based on the required skills from a set of experts denoted by (X) which have a set of skills [31]. The solution for TFP requires that members of (X') not only meet the skill requirements of the task but can also work effectively together as a team. To measure the effectiveness of the team, communication cost incurred by the subgraph in G that only involves (X') is used. Whereas, a set of required skills is a subset of a set of the total number of skills. Link prediction predicts missing links in current networks and new or dissolution links in future networks [47]. Community detection can be defined as finding nodes with the tendency of similar tastes, choices, and prefer-

ences to get associated in a social network leading to the formation of virtual clusters or communities [10]. Communities undergo a transition that can be traced as migration between communities. Collaborative recommendation identifies users whose tastes are similar to those of the given user and recommends items they have liked [8].

The scope of this thesis is focused on TFP because finding a team with the set of required skills within minimum cost in a social network is a challenging task in SNA. Finding teams in real life out of billions of people is very costly.

Previous studies about TFP aimed at finding a team and measure its performance based on the communication cost with sum of distance, communication cost with diameter, steiner tree method and minimum spanning tree method. They used one or more than one parameters for nodes network graph such as communication cost, personnel cost, load balancing and expertise level with non-knowledge based approaches in [5], [26], [27], [31], [34] and [37]. However, Genetic Algorithms are used recently in [7], [6], [23] and [30]. Further, knowledge-based approaches such as Cultural Algorithm and Genetic Algorithm are used to solve the problem in the most recent research papers such as [41] and [42].

Some practical applications of TFP in SNA are Yahoo! Answers [1], LinkedIn, Slashdot [18], GitHub [23], BitBucket[23], Kaggle and DBLP [41] [42].

1.2 Thesis Motivation

TFP in social networks is gaining importance in the fields of data mining and social network analysis.

TFP is NP-Hard problem and as per best of our knowledge, it means no optimal solution has been discovered to solve problem team formation in social networks. By solving the problem of team formation, we can reduce computational cost as well as a economic cost for finding a team from a huge and complex social network.

The motivation for this research is to find the better solution with evolutionary computation approach to finding the best team with minimum cost based on sum of distance function. There are so many approaches to test and compare the results for TFP such as Random methods and Exact algorithm. Random methods always select the team of experts randomly from the set of the experts which experts has the lowest communication cost or edge weight between them. On the other hand, Exact algorithm calculates the communication cost using exhaustive search [41]. Exact algorithm search is exponential and can take months for the bigger set of required skills. Hence, it is not a feasible solution in reality. But, evolutionary computation is a more suitable approach for finding near-optimal solutions by harnessing the knowledge within the network. In addition to this, research also used greedy algorithms to solve the problem. Greedy algorithm methods can find the solution, but their performance is lower than GA and CA [41]. Moreover, greedy algorithms produce locally optimal solutions rather than the globally optimal solution.

WSCAN-TFP algorithm worked faster than the evolutionary algorithms counterparts, but it is of lower performance compared to CAs and GAs [41]. So, the motivation for this thesis work is to develop the new hybrid approach with the help of combining modified WSCAN-TFP for social networks, which has less processing time (faster) and evolutionary computation algorithms (uses knowledge of network to harness better quality of the solution) to improve the overall quality of the solution

based on criteria explained above.

For the quality of the individual solution with proposed heuristics, we define multiple quality criteria based on communication cost (CC), average fitness score (top n-teams) and average processing time. We utilize the advantage of the genetic algorithm, cultural algorithm, modified WSCAN-TFP algorithm and schema theorem to find a method to solve the problem with the expectation of better quality individual solution.

The reason behind using WSCAN-TFP is that clustering algorithms are proved to be successful in another type of networking problems. Moreover, TFP in the social network is similar to the network problems. We can take advantage of a WSCAN-TFP algorithm based on structural similarity that it shows promising results on social network graph for TFP.

Optimization is finding the best result by maximizing the desired factors and minimizing the undesired ones. Optimization problems are the problems to find the best solution out of all the feasible solutions [12]. The optimization problem is applied to a wide range of areas like energy utilization, supply chain management, job scheduling, solving mathematical problems and much more [36]. Team Formation Problem is one of the optimization problems. Evolutionary Computation (EC) algorithms proved to be successful in optimization problems. EA optimizes the problem efficiently as it contains the search space and searches for the best possible solution in it [45]. The solutions can be either near optimal or optimal [36].

In addition to this, various non-knowledge based approaches have been proposed and used to solve the team formation problem. However, 100% quality of individual

solution hasn't been achieved yet and more research is needed to utilize the benefits of EC.

1.3 Thesis Statement

The objective of this thesis/research is to find out a group/a team of experts in a network that covers all skills from a set required skills necessary to complete a project and also minimizes the communication cost between team members. According to previous authors team with less communication cost (sum of distance) is believed to perform in an effective manner [31]. We are trying to find or form a better quality team (term "Team" is used for in TFP as an output (solution). However term "individual" is used in Evolutionary Algorithms as an output (solution)) with the help of Evolutionary Computation concepts such as Schema theorem explained in detail later on in chapter 2 and chapter 3.

We measure the quality of individual solution with the fitness function. The fitness function, we defined it later on in definition 24 and equation 3.10 as a sum of distance or communication cost and based on this fitness function $F(x)$ we calculate communication cost (CC) with a sum of distance function and Average Fitness Score (AFS) for n top-teams. We expect to see low fitness score (fitter solution) and better speed with our proposed strategies.

1.4 Thesis Contribution

This thesis contributes two novel knowledge-based approaches to solve the TFP. The first one is named as Hybrid Genetic Algorithm (HGA) and the second one is called as a Hybrid Cultural Algorithm (HCA).

In addition to this contribution, it also includes a non-knowledge based approach (a clustering method based on structural similarity) WSCAN-TFP to solve the team formation problem. Moreover, the clustering method based on structural similarity is first time used on social network and TFP in paper [41].

Moreover, to measure the quality of the individual solution, we defined quality criteria with a couple of performance measurements. We consider various performance measurements and these are communication cost(CC) with a sum of distance function, average fitness score(AFS) for to n teams and average processing time of the novel approach Hybrid Cultural Algorithm (HCA) and Hybrid Genetic Algorithm (HGA) mentioned above. We also implemented the Cultural Algorithm (CA) and Genetic Algorithm (GA). However, the DBLP dataset with 50K and 100K nodes network are used as a case study to measure the performance of all five different strategies. Later, we compare the results of all five strategies by calculating percentage difference (improvement) in the fitness of the solution. We also conducted regression analyses for HCA by increasing the size of the set of required skills to more than 1000 skills and 2000 skills.

These five strategies are as follow:

- WSCAN-TFP- strategy 1 (S1)
- Genetic Algorithm (GA)- strategy 2 (S2)
- Cultural Algorithm (CA)- strategy 3 (S3)
- Hybrid Genetic Algorithm (HGA)- strategy 4 (S4)
- Hybrid Cultural Algorithm (HCA)- strategy 5 (S5)

Moreover, HGA and HCA are able finds multiple teams/solutions. To calculate the average fitness of multiple teams, we are using average fitness score based on fitness function. But, fitness decreases as Fitness Score increases. However, we used Average Fitness Score to calculate fitness based on fitness score of multiple teams (top n-teams) found as a potential solution to the problem. We can choose a team with minimum fitness score, but if that team is not able to work in the future for some unknown reasons. We can choose the second-best team to replace it.

1.5 Thesis Organization

The rest of the thesis/research work is organized in the following manner.

In chapter II, we discuss related work/literature review in the field of team formation problem (TFP) in a social network, SCAN, WSCAN, Evolutionary Computation (EC).

In chapter III, we introduce our proposed approach which makes it possible to utilize clustering based on structural similarity to reduce the search space and utilize the advantage of evolutionary methods.

Chapter IV, we explain our experimental setup. This chapter also presents the experimental results and provides their analysis. We discuss the technical aspects of our experimental setup.

In Chapter V, we are discussing the proposed approach and provides the limitations of the proposed approach.

Chapter VI, concludes the research, explains insights received during the work and sets up the field of opportunities for the future work.

Chapter 2

Related Work and Literature Review

This chapter consists of all the related work used for the building of the fundamental concepts, developing the framework and architecture of our thesis. In this chapter, we explain the literature related to Team Formation Problem(TFP), SCAN and its variants, Evolutionary Algorithms such as Cultural Algorithm and Genetic Algorithm, and Schema Theorem by John Holland.

2.1 Social Networks

Social networks are the popular way to model the interactions among the people in a group or community as per described by the John Scott in [40] book. They can be visualized as graphs, where a vertex corresponds to a person or an expert in some group, and an edge represents some form of association/relationship between the corresponding persons [40]. The associations are usually driven by mutual interests that are intrinsic to any group or a community. John Scott in [40] describes a Social Networks as a set of nodes tied together by the set of relations (edges) between them

and these social networks follow a complex pattern and form a complex system of vertices and connections (edges) between them. A social network can be represented by a weighted or unweighted graph [40]. Where, $G = V, E$ represents an unweighted graph where V is the set of vertices (actors) and E is the set of edges (relations) [40] .

Moreover, the author writes that the social networks are ubiquitous and can be created from various disciplines such as Sociology, Twitter friendship, LinkedIn profile, protein network, etc. in [40]. Social networks are different from simple graphs. However, these networks look the same as graphs, but it satisfies some characteristic properties such as path distance (six degrees of separation), degree distribution and clustered coefficient [40]. Development social network thinking can be traced back to relational and structural approached to social analysis that developed in classical sociology. However, some approaches to sociology and anthropology used the idea of culture and cultural formation to demonstrate and explain social feelings, social patterns, social behavior, and other social causes stressed the physical environment.

Furthermore, an important strand of social through only focuses on actual patterns of interaction and interconnection through which individuals and social groups are related to each other. In some cases, it is described as a social organism or social system. In other cases, greater attention was given to face-to-face encounters through which individuals relate to each other and constantly refigure through the actions of these individuals. Frigyes Karinthy was the very first person who introduces the concept of six degrees of separations in 1929, two randomly selected people in the world are six steps away from each other. Stanley Milgram in 1960 experimented to find the average path length and found it as 5.9. Moreover, degree distribution is another prominent characteristic of the social network. It can be defined as a probability of

the number of connection of a node with other nodes over the complete network. In 1965, Derek de Solla Price found that complex network had a heavy-tailed distribution following a power law distribution. But in 1999, Albert-Lszl Barabsi et al. said that some nodes had many more connections than others, called the hub and they used the term "scale-free network" [9].

In addition to it, social networks as a complex network have another property called as clustering coefficient[40]. Clustering coefficient is the tendency of nodes clustering together and highlight the significance of the number of the triangles of the network. By calculating local and global clustering coefficient, we can have an idea of a node how likely to tie together with others and how tightly overall network be together respectively [40].

2.2 Social Network Analysis

Social Networks Analysis (SNA) can be simply defined as the in-depth analysis of social network structure, the tendency toward the time, the pattern of relationship with social actors and the available data along with them [40]. Since social networks are formed mostly with our environmental structure, researching on its primary measures such as closeness centrality, betweenness centrality, degree centrality, diameter, etc. [40] will provide more powerful results which would be an innovative change in the world [40]. To analyze a social network, we need to convert it into the graph with nodes and edges, where nodes are social actors (can be a person, organization or any other) and edges are the relationship between them [40]. The graph can either be weighted or unweighted (Weight mostly decided based on the similarity of two nodes, the distance between them or frequent relationship) [40]. At the same time, it can

be either directed or undirected to [40]. Therefore, social network analysis uses the graph theory concepts.

Another essential characteristic of the social network is that it shows dynamic behavior. The complex networks are said to be dynamic networks when their topology changes over times. Real-world social networks, however, are not always static. In fact, most popular social sites in reality (such as Facebook, Twitter, and LinkedIn) evolve heavily and witness a rapid expansion regarding size and space over time. The rapid and unpredictable changes of topological structure of the complex networks make extremely complicated and yet challenging problems. For example, it helps to analyze the spread of diseases [19], to detect terrorist activities [46], to observe dynamic co-authorship networks [29] and many more researches applications in the real-world.

2.2.1 Application of SNA

In recent years, SNA has been used in various disciplines in business, academics, politics, health care and daily life activities [40]. It is most commonly applied to help to improve the effectiveness and efficiency of decision-making processes [40].Applications of SNA are used in field such as Business, Law enforcement agencies (and the army), Social Network Sites, Civil society organizations, Politics, Spread of Diseases, Health care.

2.2.2 Various SNA Problems

Author of [40] writes that Social Network Analysis (SNA) deal with different issues. Few of them described by the author in [40] are the very hot trend in SNA research:

Team Formation Problem, Link Prediction, Leadership Detection, Community Detection, Migration Between Communities, Sentimental Analysis, Collaborative Recommendation, Influence Analysis, Fraud Detection.

TFP is finding an optimal solution to find a team to complete a project P, which has a set of k required skills denoted by as criteria. The team is selected based on the required skills from a set of experts denoted by which have a set of skills denoted by [31]. Whereas, a set of required skills is a subset of a set of the total number of skills. Link prediction predicts missing links in current networks and new or dissolution links in future networks[47]. Community detection can be defined as a finding people with the tendency of similar tastes, choices, and preferences to get associated in a social network leads to the formation of virtual clusters or communities [10]. Communities transit and this transition can be traced as migration between communities. Collaborative Recommendation identifies users whose tastes are similar to those of the given user and recommends items they have liked [8].

However, this thesis is focused on TFP, because finding a team with the set of required skills within minimum cost in a social network is a challenging task in SNA. Finding teams in real life out of a billions of people is very costly and not feasible.

2.3 Team Formation Problem (TFP)

Lappas et al. (2009) in his work [31] proposed two communication cost functions and used Rarest first and Enhanced Steiner algorithm to discover the team of experts from a social network. Later, the problem was approached with generalized enhanced Steiner algorithm by Li and Shan and extended the work of research paper [31] in [32].

Another method was proposed by Kargar and An (2011) who introduced a team with the leader that minimize leader distance function and produce the top-k team in their work [26]. Moreover, Gajewar and Sharma (2012) presented another cost function based on density in the research paper [21]. Anagnostopoulos et al. (2010) ignored the communication cost among experts while dealing with multiple projects to minimize the maximum load of experts in his work [4]. Then again, Anagnostopoulos et al. (2012) experimented in [5] by minimizing both load balance and communication cost.

Kargar et al. (2012) assumed in [27] that every expert is associated with a cost to perform an assumed task in a given project. By using the trade-off parameter in [27], they combined two objective functions into one. Moreover, Kargar et al. (2013) found the best team in [28] by minimizing the communication cost under given personal cost budget. To solve this problem Kargar et al. (2013) found the set of Pareto teams in [28]. Li et al. (2015) solved the problem of a team member becomes unavailable by finding a replacement in [32]. Awal et al. (2014) proposed to find a team of experts in the social network in his research [7] using collective intelligence index. They used a random expert in [7] to optimize communication cost and expert level with implementation of the general genetic algorithm (GA).

Wi et al. (2009) evaluated two different selection methods in [50] to choose team members and project managers in their work. They studied team formation in an organization by using GA and the knowledge-based competence score of candidates for a certain project in [50]. Reynolds in 1994 introduced cultural algorithms in his work [38] . However, it was never used for team formation problem before. But Recently, the authors applied cultural algorithms to find the better optimal solution in [42] by extracting knowledge from the initial population and update to the next population. It was a little better result compare to Genetic and Greedy Algorithms. Later, we proposed WSCAN-TFP algorithm to solve team formation problem in [41]

and compared its results with other algorithms. Genetic algorithm, Cultural algorithm, Random algorithm, Greedy algorithm and Exact algorithm results were used as a baseline for comparison in [41].

Now, we will discuss some of the research paper mentioned above in detail.

Lappas et al. (2009) tried to solve the problem of team formation with a social network. They described team formation problem [31] as Given a task T, a pool of individuals X with different skills, and a social network G that captures the compatibility among these individuals, we study the problem of finding X', a subset of X, to perform the task. Following are definition from paper [31] Finding a team of experts in social networks by Lappas et al. (2009).

Definition 1. ***(Problem definition:)*** *Given the set of n individuals* $X = f(1.....n)$*, a graph* $G = (X, E)$*, and task* T*, and* $X' \subseteq X$ *, so that* $C(X'; T) = T$*, and the communication cost* $Cc(X')$ *is minimized ([31]).*

In [31] paper, author focused on two instantiations of communication cost. However, communication cost definition is not elaborated in the problem definition above to make the definition more generalized.

First instantiation: Lappas et al. (2009) used diameter communication cost of X'; denoted by $Cc - R(X')$ in this paper.

Diameter of a graph: In general definition, the diameter of a graph can be described as a largest shortest path in between any two nodes. However, the authors described it as follow to make more suitable for team formation problem [31].

Diameter (R): Given graph $G = (X, E)$, and a set of individuals $X' \subseteq X$, diameter communication cost of X', to be the diameter of the subgraph $G[X']$ [31].

Second instantiation: Lappas et al. (2009) used minimum spanning tree communication cost of X'; denoted by $Cc - MST(X')$ in [31] paper.

Cost of spanning tree: In general definition, cost of spanning tree can be described as the cost of a spanning tree is simply the sum of the weights of its edges [31].

Minimum spanning tree (MST): Given graph $G = (X, E)$, and a set of individuals $X' \subseteq X$, minimum spanning tree communication cost of X', to be the minimum spanning tree communication cost of the subgraph $G[X']$ [31].

Authors in paper [31] called Team formation problem with two communication cost functions as mentioned above.

$communicationfunction$	$ProblemTF(TeamFormation)$
$Cc - R$	$Diameter - TF; Cc - R(X')$
$Cc - MST$	$MST - TF; Cc - MST(X')$

Authors observed that RarestFirst, GreedyDiameter, EnhancedSteiner, and GreedyMST produce approximately the same number of disconnected teams in the paper [31].

The author of [31] claimed to address the problem of forming a team of skilled individuals to perform a given task while minimizing the communication cost among the members of the team. The author in [31] claimed that teams formed by their algorithms on a set of real tasks. Authors observed that CoverSteiner and Greedy-

Cover often fail to and a connected team, even in cases where such a team actually exists. The results in [31] indicate that, although GreedyCover produces teams of small size, the members of this team cannot communicate efficiently.

Anagnostopoulos et al. (2010) in [4] paper presented a general framework for task assignment problems. Further, he provided a formal treatment on how to represent teams and tasks. However, he proposed alternative functions for measuring the fitness of a team performing a task and discussed desirable properties of those functions in [4] and he also provided algorithms with provable approximation guarantees, as well as lower bounds in [4].

In [4] paper, Anagnostopoulos et al. (2010) proposed the algorithmic tool to help people collaborate efficiently. The author defines this problem in [4] as to assign tasks J^j from a set J to teams Q_j, which are subsets of people P so that teams are fit for their tasks and the assignment is fair to people.

$Symbol$	$Definition$
$Tasks(J)$	$J = J^j; j = 1, 2, ..., k$
$Skills$	$J^j \in S.$
$People(P^j)$	$P = P^j; j = 1, 2, ..., n$
$Teams(Q^(j))$	$Q^j \subseteq P$
$Scoring function(s)$	$s(q, J), s(,) \in [0, 1]$
$LoadL(p)$	$L(P) = \|J; P \in Q^j\|$

In the table above author used the set of tasks (or jobs) J, which arrived off-line or on-line scenario and needed to be assigned to a team of experts in [4]. Each task requires a set of skills. Skill space (S) that is the possible way of combining skills to complete a task. Therefore, $J^j \in S$. However, set of people (or experts) have written

down as $P = P^j; j = 1, 2, ..., n$ [4].People possess a set of skills and their profile is represented by a point in the skill space: $P^j \in S$ and tasks for individual P_i^j. Measurement the performance of the team for the task using a scoring function $s(q, J)$ [4]. It measures complete failure as a 0 and complete success as 1. $s(,) \in [0, 1]$. Moreover, It is needed to assign a task to a team to be completed [4]. Hence $Q^j \subseteq P$ assigned to j^th task. Moreover, the skills of each team represented by skill space; $Q^j \in S$. load $L_{(p)}$ of a person p, which is the number of tasks in which particular individual participates. It is described as $L(P) = |J; P \in Q^j$[4].

To explain the concept in the real world, let's take an example; We want to assign team members for an operation. For instance, this operation needs a nurse, a surgeon, and an anesthetist to complete the operation. However, the operation can not take place if any of expert with a specific skill is not found. This is the simplest general example we are taking to make the concept easy to understand.

In paper [4] algorithm picks the team of minimum size among those that have all of the required skills for specific given task. In paper [4] heuristic tries to minimize the size of the teams, it does not keep track of the work done so far and can overload the few experts that possess most of the skills. The author claimed in [4] to have better results in both theoretical and experimental with the greedy methods for an on-line scenario. That can be effective in practice, as long as they consider both team sizes and workload of members.

In [26], the authors focused on the issue of finding a group of specialists from an informal community. Given a task whose fulfillment requires an arrangement of aptitudes, Author located an arrangement of specialists that together have most of the required aptitudes and furthermore have the negligible correspondence cost among

them in [26]. However, author proposed two correspondence cost capacities intended for two sorts of correspondence structures. They demonstrated in [26] that the issue of finding the group of specialists that limits one of the proposed cost capacities is NP-hard. In this way, an estimation calculation with an estimation proportion of two is planned by authors in [26]. They presented the issue of finding a group of specialists with a pioneer/leader in [26]. The leader considered in [26] as in charge of checking and organizing the venture, and in this way, an alternate correspondence cost work is utilized as a part of this issue.

Shortcomings of previous work in this area only found one single best answer. The author presented in his paper [27] that two procedures that enumerate top-k teams of experts with or without a leader in polynomial delay. Authors of [27] introduced two new cost functions. They consider two types of communication structures/functions within a team. They supposed that each required skill corresponds to a task in the project [27]. However, in the first communication structure, the experts for each pair/set of required skills need to communicate with each other to complete the corresponding tasks in [27]. For such a structure, author in [27] defined a cost function that they called it Sum of Distances.

Moreover, to calculate the communication cost of a team using the sum of the shortest distances between the experts for each pair of skills used by the author in [27]. In the second type of communication structure/function author mentioned in [27] that a leader needs to communicate with each team member to track and coordinate the project. For such a structure author defined by [27] gave a cost function and called it Leader Distance. However, it computed the sum of the shortest distances between the leader and each skill holder in the team.

Author of paper [27] used the Exact algorithm to find a team with or without the

leader. The author claimed to propose an effective and scalable method on large real data. The author claimed in [27] to have the following contribution in this paper.

The author proposed two new functions in [27] for measuring the communication cost of a team of experts in social networks. 1. Sum of Distances 2. Leader Distance Author proved in [27] that the problem of finding a team of experts that minimizes the Sum of Distances function is NP-hard. The author introduced in [27] that the problem of finding a team of experts with a leader that minimizes the Leader Distance function. The author in [27], enumerated top-k teams of experts with or without a leader in polynomial delay.

However, In [28] paper, Author extended their previous a work of previous paper [27] in this paper. To enhance, they defined a new combined cost function in [28] paper, which is based on the linear combination of the objectives 1. communication and 2. personnel costs. They showed that the problem of minimizing the combined cost function is an NP-hard problem in team formation problem in [28] paper. Therefore, an approximation algorithm is used in [28] paper to solve the problem.

The author in [28] proposed four algorithms for finding a team of experts in a social network that minimizes both the communication cost and the personal cost of the team. The author in [28] used an approximation algorithm with a provable performance bound as a rst algorithm and for the other three algorithms use heuristics to nd sub-optimal solutions. The author claimed to have better results in [28]. The author said that proposed methods in [28] are much faster than the Random and Exact methods and Random method has the highest cost.

In [37], Author defined the problem as Given a task T, a set of experts V with

multiple skills and a social network $G(V, W)$ reflecting the compatibility among the experts, team formation is the problem of identifying a team $C \subseteq V$ that is both competent in performing the task T and compatible in working together.

In [37] paper, they proposed a new approach based on an older approach called Densest Subgraph Problem (DSP) with cardinality constraints. However, this is an NP-hard problem, but it has many applications in real-world social network analysis. They proposed the new method in [37] that can solve (approximately) the Generalized Densest Subgraph Problem (GDSP). Experiments conducted by the author in [37] shows that proposed formulation GDSP is useful in modeling a broader range of team formation problem and it produced more coherent and compact teams of high quality.

Author experimented using DBLP data in [37], they choose four fields of computer science stream.

Fields

- Databases (DB)
- Theory (T)
- Data Mining (DM)
- Artificial Intelligence (AI)

Conferences that author considered in [37] for each field are given as follow:

Conferences

- DB = SIGMOD, VLDB, ICDE, ICDT, PODS
- T = SODA, FOCS, STOC, STACS, ICALP, ESA
- DM = WWW, KDD, SDM, PKDD, ICDM, WSDM
- AI = IJCAI, NIPS, ICML, COLT, UAI, CVPR

Selected skills

- A = DB, T, DM, AI

Any author who possesses at least three publications in any of the above 23 conferences was considered expert for the experiment in [37]. In this experiment using DBLP co-author data a graph was generated, where, a vertex corresponded to an expert and an edge between two experts indicate prior collaboration between them. The weight of the edge is the number of shared publications considered as an edge weight for the experiment in [37].

The author in [37] claimed to find qualitatively better teams that were more compact and have higher densities than those found by the greedy method. However, linear programming relaxation not only allowed to check the solution quality but also provided a good starting point for our non-convex method [37]. The author tested results in [37] with the greedy algorithm and claimed to get better results. However, he also mentioned a potential downside of a density-based approach is that does not guarantee connected components. A further extension of his approach could aim at incorporating connectedness or a relaxed version of it as an additional constraint [37]. The author claimed to find the optimal solution in [37] with the implementation of the greedy algorithm. However, the greedy algorithm provides an optimal solution

for the above example, but it may not provide an optimal solution for all problems [37].

Authors in [20] presented a mathematical framework for treating the Team Formation Problem by explicitly incorporating Social Structure (TFP-SS). The formulation of this mathematical framework relied on modern social network analysis theories and metrics. Moreover, in [20] paper to solve TFP from a given a pool of individuals, the TFP-SS assigned them to teams to achieve an optimal structure of individual attributes and social relations within the teams. The author in [20], explored TFP-SS instances with measures based on such network structures as edges, full dyads, triplets, k-stars, etc., in undirected and directed networks.

Shortcomings of previous research papers according to the author in [20] were solving most problems and addressed by observations, experiments, and basic statistical methods. The author in [20] justified the use of mathematical programming and optimization techniques in the area of social science. The author used the various graph-based diagram (mathematical approach) for social network theories and proposed LK-TFP algorithm in [20] paper. Author used LinKernighan-TFP (LK-TFP) heuristic in [20] that performs variable-depth neighborhood search. In [20] paper, the author described LK-TFP as a tree search procedure and made the contribution as LinKernighan TFP (LK-TFP) algorithm for solving TFP-SS, based on variable depth-first neighborhood search.LK-TFP traverses the tree to arrive at such a transitive solution that improves the objective function [20].

Author of [20] paper described tree traversal in detail: the root node represents a feasible solution for TFP-SS. However, internal tree nodes (at $slevels \in S$) represent solutions resulting from s moves performed on the root solution [20]. Further, it moves to leaf nodes. It will not remove the last node in every branch, that means

one individual with minimum quality to be a team member [20].

The author in [20] paper claimed to Identify good branches of the tree and avoids visiting too many non-improving solutions by cutting off the search space. The author claimed to have the high-quality solution with use of mathematical programming and optimization techniques in the area of social science in [20] paper.

The author in [52] formulated three ranking objectives to optimize communication cost, skill holder authority, connector authority and combinations of them. The author in [52] paper proved that optimizing these objectives is an NP-hard problem. Moreover, the author jointly considered communication cost and expert authority to find out the pureto optimal teams. The author presented an algorithm in [52] to optimize communication cost over an expert network G and a transformation that moves authority (node weights) onto the edges of a new graph G' and proved that their algorithm also optimized the other objectives over G'.
They performed a comprehensive evaluation using the DBLP dataset to confirm the effectiveness and efficiency of their approach in [52].

The table shows Team Formation Problem (TFP) with the various approach in Social Network Analysis (SNA).

Author [year]	$CC-R$	$CC-Steiner$	$CC-SD$	$CC-LD$	Algorithm/ Approach
Lappas et al., 2009 [31]	Yes	–	–	–	$RarestFirst$
Lappas et al., 2009 [31]	–	Yes	–	–	$EnSteiner$
Kargar et al., 2011 [26]	–	–	Yes	–	$MinSD$
Kargar et al., 2011 [26]	–	–	–	Yes	$MinLD$
Kargar et al., 2012 [27]	–	–	Yes	–	$MCC, ItReplace$
Majumder et al., 2012 [34]	Yes	–	–	–	$MinDiaSol$
Majumder et al., 2012 [34]	–	Yes	–	–	$MinAggrSol$
Anagnostopoulos et al., 2012 [5]	Yes	–	–	–	$LBRadius$
Anagnostopoulos et al., 2012 [5]	–	Yes	–	–	$LBSteiner$

CC-R (communication cost with diameter function): Author in [31], [34] and [5] used the diameter of the team (diameter is longest shortest distance) to calculate communication cost. **CC-Steiner** (communication cost with Steiner tree): Author in [31], [34] and [5] used Steiner tree method to calculate communication cost for different teams. **CC-SD** (communication cost on the sum of distance function) : Author in [26] and [27] used sum of distance function to calculate communication cost for the team. Where sum of distance function is a summation of all edge weights connecting team member nodes.

2.4 Graph Clustering

Clustering can be defined as grouping together one type of elements into a group. With clustering, we can produce as many as groups based on characteristics of the elements under consideration. Moreover, Network clustering (or graph partitioning) is an important task for the discovery of underlying structures in networks [51].

Methods of Clustering can be divided into following categories:

- Hierarchical based clustering
- Density-based clustering

- Partitioning based clustering
- Grid-based clustering

2.4.1 Density Based clustering

In the density-based clustering method, grouping is done based on highly connected nodes in a graph. All highly connected vertices are identified and made a cluster. Example: DBSCAN (Density-Based Spatial Clustering of Applications with Noise), AHSCAN, DHSCAN, SCAN.

DBSCAN algorithm was proposed for clustering spatial data with noise. Because of its unique features, this algorithm became rapidly popular in various field and application of this algorithm includes in the field of science as grouping spatial civil infrastructure network, chemistry, spectroscopy, medical diagnosis based on medical images (brain atrophy, skin lesions) and social science (pheromone data) [44]. It can also be applied on remote sensing to perform segregation of 3D images.

The disadvantage of DBSCAN algorithm was that it failed to determine when the border elements of two clustering are relatively too close. Later, Structural Clustering Algorithm for Networks was proposed by [51]. This algorithm can cluster densely connected as well as weakly connected nodes with hubs. They describe hubs as those nodes which are connected to more than one cluster.

2.5 Unweighted graph clustering with SCAN

In the research paper [51], the author proposed a new method to cluster a network graph based on structural similarity. This method was used to cluster unweighted and

undirected graph based on structural similarity. Hence, author named it as Structural Clustering Algorithm for Networks(SCAN). It detects clusters, hubs, and outliers by using the structure and the connectivity of the vertices as clustering criteria [51]. This algorithm finds Core node out of network. The Core node is chosen based on the number of neighbors in its neighborhood are structurally similar. Later, by considering this Core node as the seed for the cluster, it builds up cluster around it. This approach divides the graph into three parts: Clusters, Hubs, and Outliers.

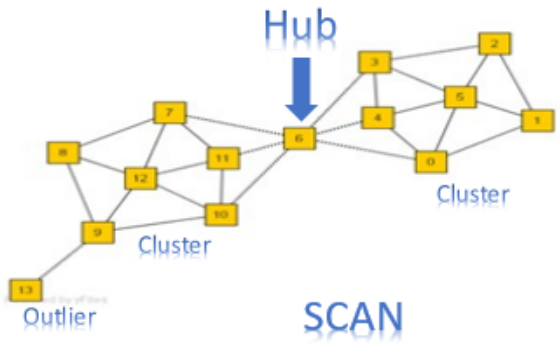


Figure 2.1: Working of SCAN in research paper [51].

Formal definitions used by the author in [51] are described below for the reference.

Definition 2. ***(Vertex structure)***

Let $v \in V$, the structure of v is defined by its neighborhood, denoted by $\tau(v)$ [51].

$$\tau(v) = u \in V \vee (u, v) \in V\} \cup \{u\} \tag{2.1}$$

Definition 3. ***(ϵ - Neighborhood)***

When a vertex shares structural similarity with enough neighbors, it becomes a nucleus or seed for a cluster[51].

$$N_\epsilon = \{u \in \tau(v) | \sigma(u, v) \geq \epsilon\} \tag{2.2}$$

Definition 4. ***(Structural similarity)***

Structural similarity of two vertices/ Experts will be large if they share a similar structure of neighbors. A minimum (threshold) value of structural similarity ϵ is introduced by this definition[51].

$$\sigma(u, v) = \frac{|\tau(u) \cap \tau(v)|}{\sqrt{|\tau(u||\tau(v|)}} \tag{2.3}$$

Definition 5. ***(Core)***

A vertex $v \in V$ is called a core with reference to ϵ and μ, if its $\epsilon - neighbourhood$ contains at least μ vertices [51]. Core vertices are a special class of vertices that have a minimum of μ neighbors with a structural similarity that exceeds the threshold ϵ [51].

$$Core_{\epsilon,\mu}(v) \leftrightarrow |N_\epsilon| \geq \mu \tag{2.4}$$

Definition 6. ***(Direct structure reachability)***

Two non-core vertices in the same cluster may not be structure reachable because the core condition may not hold for them [51].

$$DirREACH_{\epsilon,\mu} \iff Core_{\epsilon,\mu}(v) \wedge u \in N_\epsilon(v) \tag{2.5}$$

The search begins by first visiting each vertex once to find structure-connected clusters and then visiting the isolated vertices to identify them as either a hub or an outlier [51]. The pseudo code of the algorithm SCAN is presented below. SCAN

performs one pass of a network and finds all connected clusters for given parameter settings. In the beginning, all vertices are labeled as unclassified [51]. The SCAN algorithm classifies each vertex either a member of a cluster or a nonmember. For each vertex that is not yet classified, SCAN checks whether this vertex is a core [51]. If the vertex is a core, a new cluster is expanded from this vertex. Otherwise, the vertex is labeled as a non-member[51]. SCAN begins by inserting all vertices in $\epsilon - neighborhood$ of vertex v into a queue [51]. For each vertex in the queue, it computes all directly reachable vertices and inserts those vertices into the queue which are still unclassified [51]. This is repeated until the queue is empty[51].

A network is sets of vertices, representing objects, connected together by edges, representing the relationship between objects[51]. For example, a social network can be viewed as a graph where individuals are represented by vertices and the friendship between individuals are edges [48].

2.6 Weighted graph clustering with WSCAN

Every graph edge may have a positive number associated with it, which is usually called edge weight or capacity [13]. Algorithm mentioned above have one common property - it targets unweighted graphs. However, When provided with a weighted graph, any of the algorithms mentioned above will simply ignore the edge weights and will perform clustering based on structural properties of the graph. While this might be acceptable in certain cases, sometimes it is completely inadmissible [14]. To overcome this problem author of [14] research paper and [13] thesis research work, proposed a new algorithm called Weighted Structural Clustering Algorithm for Networks (WSCAN) as a solution to perform clustering in weighted graphs based on structural similarity.

Definition 7. ***(Extended Structural similarity)***

$v, u \in V$ and function below shows structural similarity in a graph with $e(u, v)$ edge weight between node v and u [14], [13].

$$\sigma(u, v) = \frac{|\tau(u) \cap \tau(v)|}{\sqrt{|\tau(u||\tau(v|)}} e(u, v) \tag{2.6}$$

In equation 2.6 shows that extended structural similarity of two vertices will be large if they share a similar structure of neighbors. A minimum (threshold) value of structural similarity ϵ is introduced by this definition [14], [13].

2.7 Evolutionary Computation

Evolutionary Computation (EC) is sub-branch of artificial intelligence (AI), which is used for metaheuristic and stochastic optimization of complex problems. It is the set of evolutionary algorithms which are inspired by the biological model of evolution [45]. The algorithms that come under this section adopt Darwin's principles of Evolution; hence, they are called Evolutionary Algorithms [45]. Technically speaking these algorithms can be considered as Global optimization problems according to Kybernetes (1998) mentioned in research work [45].

2.7.1 Evolutionary Algorithms

Evolutionary algorithms (EA) has been used widely by the researchers to solve the optimization problems. EA optimizes the problem efficiently as it contains the search space and searches for the best possible solution in it [45]. The solutions can be

either near optimal or optimal. EA allows the exploration and exploitation of the search space. Exploration helps to search the whole space and exploitation helps the solution to mutate and generate offspring [36]. Evolutionary algorithms (EA) is a subset of EC, and hence they are also considered as optimization algorithms. The common underlying concept in each evolutionary algorithm is the same. Given a set of the population under environmental pressure causes natural selection [36]. The function measures the of the candidates, and the better candidates survive for the next generation, discarding the worst ones [36]. Evolution of every individual is carried out by applying mutation and recombination operators on it [36].There are various algorithms which come under EC, such as:

1. Genetic Algorithms
2. Cultural Algorithms
3. Differential Evolution
4. Particle Swarm Optimization
5. Ant Colony Optimization Algorithm

2.7.2 Genetic Algorithm

Genetic Algorithms (GA) are a subset of EA; hence they are population-based evolutionary algorithms. Genetic Algorithms were introduced by Holland [25] but became popular after the works of Goldberg [11]. Genetic Algorithm is prominently used to resolve the search related and other optimization problems. They are very helpful to search solution, even when very less is known about the domain [36]. Genetic Algorithm is consisting of a group of individuals known as population [36]. However, these individuals are used to search the optimal solution within the specified search

space [36]. An initial random population is generated over the search space and evolutionary operators like mutation, recombination and selection are applied to them [36].

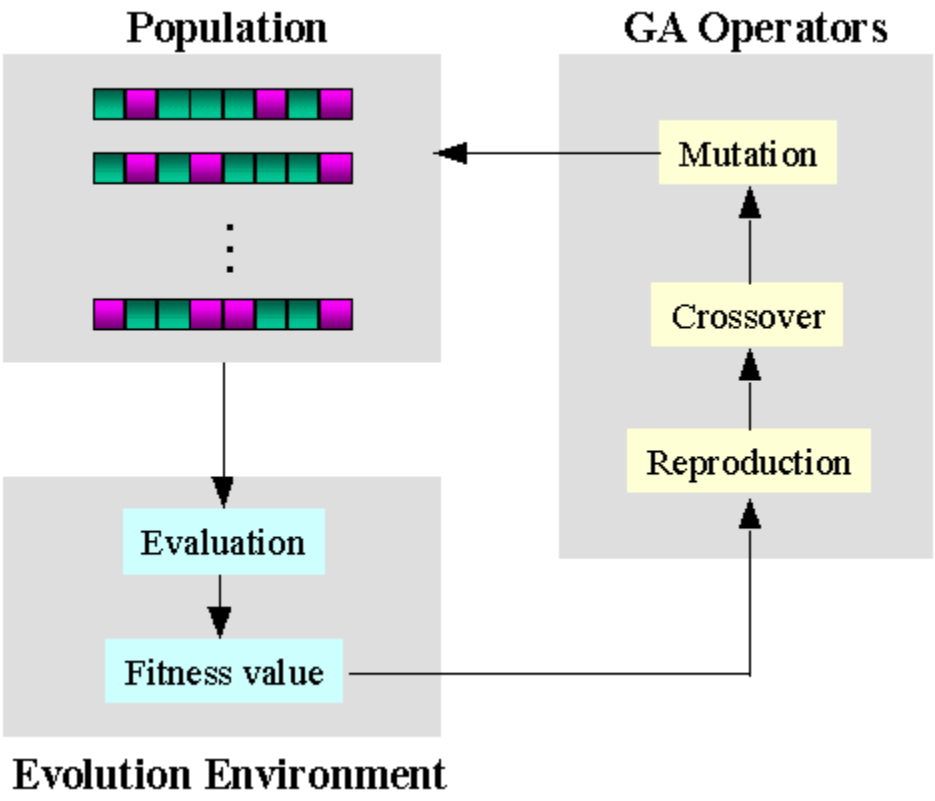


Figure 2.2: Processing of Genetic Algorithm [33]

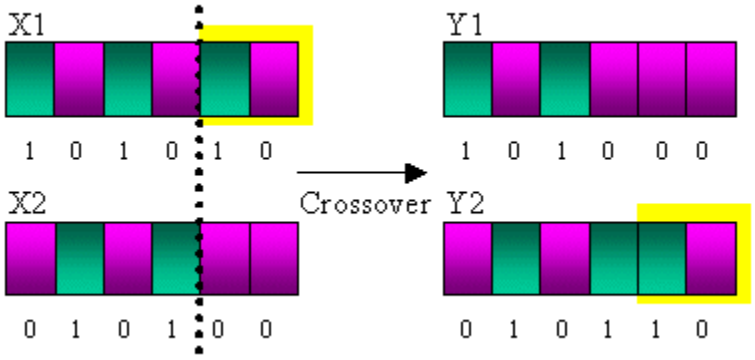


Figure 2.3: Crossover operation in Genetic Algorithm [33]

Author [year]	Algorithm/Method	Approach and Parameters
Wi et al., 2012 [50]	Genetic Algorithm	Fuzzy inference system
Dorn et al., 2010 [18]	Simulated Annealing	Expert level and communication cost
Ani et al., 2010 [6]	Genetic Algorithm	Balanced programming skills among team members
Agustin et al., 2012 [2]	Genetic Algorithm	Parallel hybrid model
Awal et al., 2014 [7]	Genetic Algorithm	Collective Intelligence Index, communication cost
Han et al., 2017 [23]	Genetic Algorithm	Communication cost and geographical distance
Selvarajah et al., 2017 [42]	Genetic Algorithm	Communication cost
Selvarajah et al., 2018 [41]	Genetic Algorithm	Communication cost

Table 2.1: Genetic Algorithm and Simulated Annealing with various parameters used to solve TFP

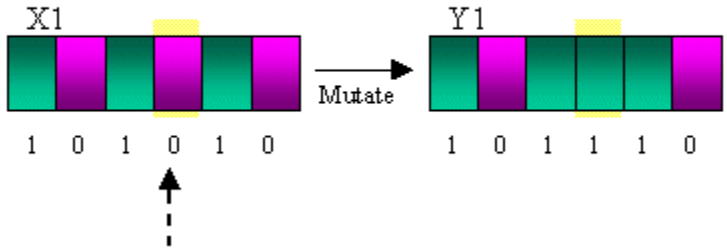


Figure 2.4: Mutation operation in Genetic Algorithm [33]

In Genetic Algorithms after each generation, the best individuals are selected for mutation, recombination, selection, and crossover [36]. The individuals also exchange knowledge among them by using these operators [36]. Genetic Algorithm is simple to code, and the population is not initialized at one point. Instead, they are spread across the search space for exploration [36]. Genetic Algorithms use mutation, crossover, and selection operator to achieve an optimal solution and enhance exploration and exploitation [36].

In [18], the author used two parameters that are Expert level and communication cost to find a team. The author evaluated team formation mechanism with a real-world dataset extracted from Slashdot in [18]. The author used Slashdot, which

is a well understood and rich data set [22] describing a large user community. The author describes that users submit information technology related news items which the editors decide to publish or not. Later, this news falls into multiple categories (i.e., subdomains) such as Linux, apple, or games [18]. A published piece of news becomes a story which all users–anonymous or logged–in can comment on [18]. These comments create a posting hierarchy. Slashdot exhibits the characteristics of a large-scale expert network [18]. The author discusses in [18] that optimal team composition that requires a trade-off between skill coverage and expert connectivity. The author claimed to demonstrate the benefit of our heuristic for finding well-connected experts that simultaneously yield a high expertise level in a social network in [18] .

The author used two parameters that are geographical distance and communication cost in [23] to find a team of experts. In [23], the author claimed the first parameter that the proposed GA based model achieves better performance with the sum of geographical proximity evaluation metric, whereas the random algorithm gets the worst. The author mentioned in the research paper that GA based model achieves better results because the GA-based model considers the sum of geographical proximity during the process of finding an optimal team in [23] paper. For the second parameter, the proposed GA-based model also achieves better performance on the sum of the communication cost evaluation metric, whereas the and random algorithm performed worst. This is because the GA-based model has a larger search space while MCC-Rare algorithm and approximation rare algorithm has a smaller one [23]. The random algorithm does not consider the sum of communication cost factor [23].

In [41], the author finds a team of the experts from a social network by taking communication cost as a parameter. The author proposed WSCAN-TFP algorithm

in paper [41]. WSCAN-TFP is a clustering method based on structural similarity. The author in [41] compared results from WSCAN-TFP with Genetic Algorithm, Cultural Algorithm, random algorithm, greedy algorithm and exact algorithm as a baseline for results. However, the author found that WSCAN-TFP is fast compared to evolutionary algorithm counterparts; But the performance of WSCAN-TFP was less than Genetic Algorithm and Cultural Algorithm in the paper [41].

2.7.3 Schema Theorem

The Schema Theorem for genetic algorithms (GA) [25] defines how useful structures in a population of strings are propagated during the evolution of a solution [49].

Formal definitions are described below for reference [25], [11].

Definition 8. *$(Schema, H)$ A schema is a subset of the space of all possible individuals for which all the genes match the template for schema H. Suppose, A denotes the alphabet of gene alleles then $A \cup *$ is the schema alphabet, where * is the wild card symbol matching any allele value [25], [11].*

Definition 9. *$(SchemaOrder, o(H))$ Schema order $o(H)$, is the number of non * genes in schema H. For example: o (* *0 * * * *) = 1*

Definition 10. *$(SchemaDefiningLength\delta(H))$ Schema Defining Length $\delta(H)$, is the distance between first and last non * gene in schema H. Example, δ (* * 0 * * * *) = 3 3 = 0*

Definition 11. ***(Selection Operators Fitness Proportional Selection)*** *Essentially all that we are attempting to model is the probability that individual e, samples schema H, or* $P(e \in H)$.

Definition 12. ***(Schema: Fitness)*** $f(x)$ *shows fitness of bit string* x *and* $f(H,t)$ *denotes average fitness of instances of the schema in the population at* t^{th} *generation*

.

$$f(H,t) = \frac{\sum_{x \in H} f(x)}{m(H,t)} \tag{2.7}$$

2.7.4 Cultural Algorithm

The main feature of cultural algorithms that distinguish them from others is employing knowledge [38]. It is a knowledge-based evolutionary algorithm. The cultural algorithm as shown in Fig. is a dual inheritance model which consists of two main spaces, population, and culture or belief space. According to the model, in each generation, a group of individuals is selected to update the belief space and the new population is generated based on the parameters which were defined in the belief space [35]. The belief space in this model by [38] acts as a global knowledge repository which is made of information about the individuals and can be used to guide the search direction [35].

Author (year)	Algorithm/Method	Approach and Parameters
Selvarajah et al., 2017 [42]	Cultural Algorithm	Communication cost
Selvarajah et al., 2018 [41]	Cultural Algorithm	Communication cost

Table 2.2: Cultural Algorithm used to solve TFP

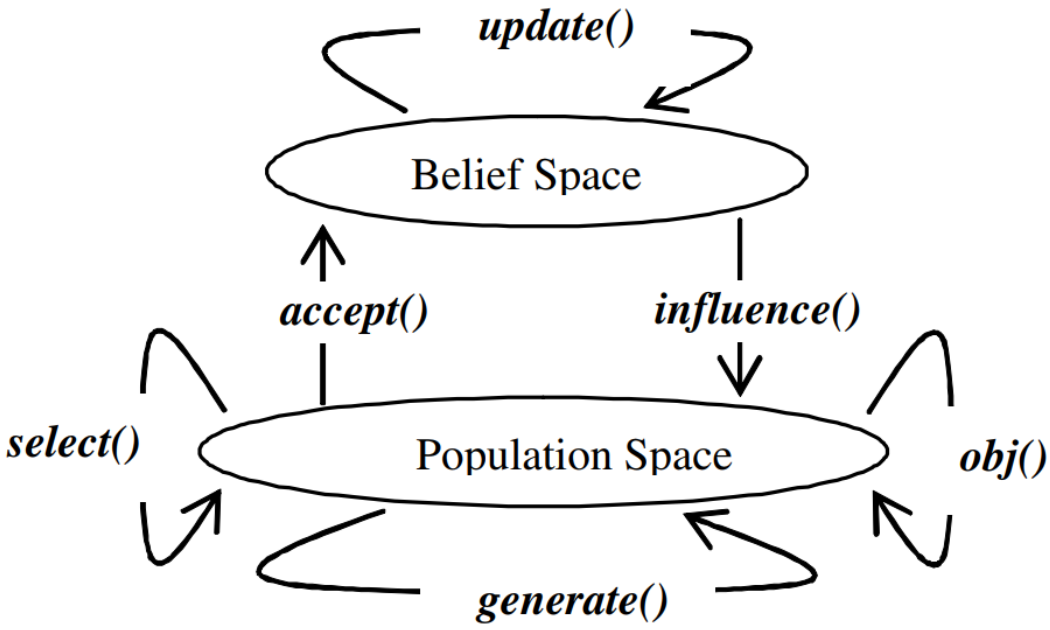


Figure 2.5: Processing of Cultural Algorithm [39]

The author in [41], used the real data set of DBLP. We have conducted experiments to get results with 50K nodes derived from the DBLP dataset. For the application of weighted SCAN function, We use the sum of distance as a communication function to calculate the weight between two experts.

To have a baseline comparison, we use random methods in [41], which always select the team of experts randomly from the set of the team which has the lowest communication cost. The Exact algorithm calculates the communication cost using an exhaustive search.

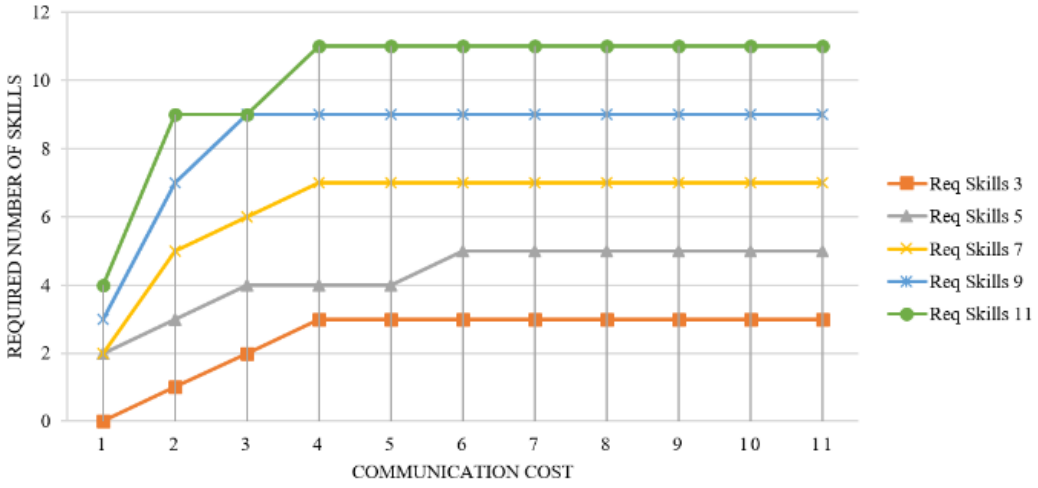


Figure 2.6: The effect of the threshold for WSCAN [41]

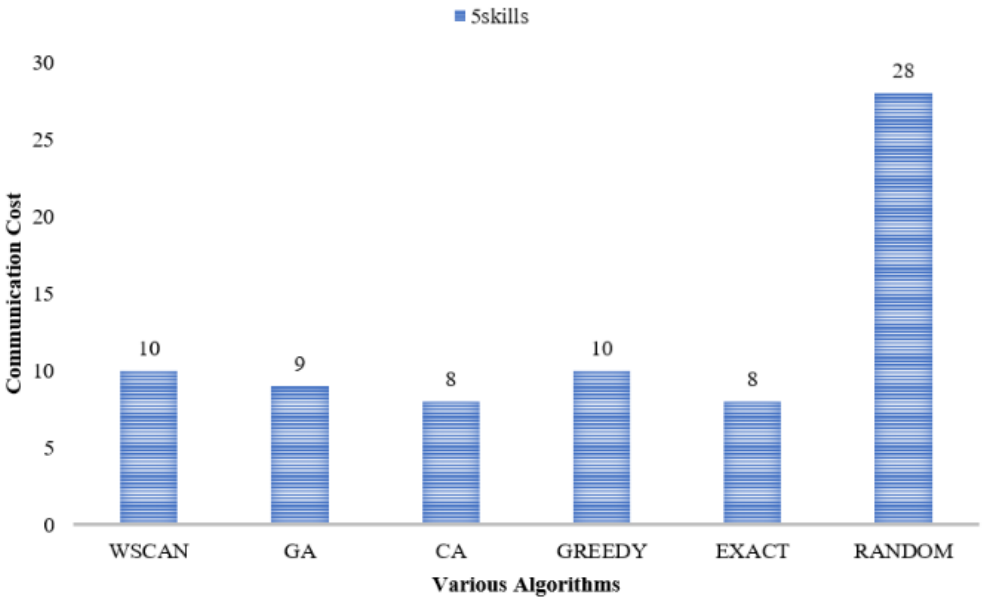


Figure 2.7: Comparison of various algorithms for TFP with WSCAN for the project require five skills [41]

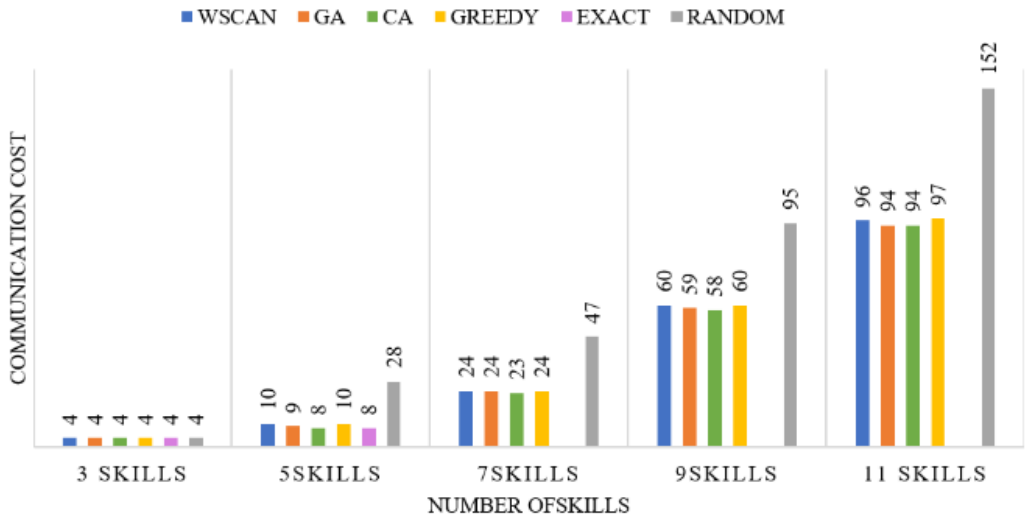


Figure 2.8: Comparison of the communication cost of a team of experts for various number of skills with different algorithms [41]

To test our algorithm on a real network, we use the DBLP[1] dataset in [41] paper, which is one of the expert's network used in [27] and [31]. The basic concept of DBLP network is, when two authors publish any paper together, they will have a connection between them. We generate the 50K nodes of equal edge weight graph with 1.0 of weight on all edges [41].

The SCAN requires threshold value to form structural similarity with neighborhood nodes [41]. Therefore we tested with the different number of skills to find them at most value as a threshold [41]. The experiment has been shown in figure 3.16 with the communication cost vs. the number of required skill graph. From this graph, we can assign a threshold of 4.0 to find the nearest neighborhood [41].

The experiment always begins by calculating communication cost from CORE expert to the neighborhood [41]. Therefore, we calculated the value of communication

[1] *http://dblp.uni-trier.de/xml/*

cost of the team of experts with required skills [41]. The figure 2.7 shows the comparison for the communication cost of a team for the required number of skills five with various algorithms. It shows approximately equal value with Greedy algorithms [41]. However, with both Cultural and Genetic algorithms, WSCAN didn't perform well. Then we examine by varying the number of required skills for a specific project as shown in the figure 2.8 [41]. However, we found that the result always follows the same findings as we saw in Figure 2.7 [41]. Importantly, the runtime of the SCAN was less than the all other algorithms [41].

Chapter 3

Proposed Approach

We have developed the number of strategies of increasing complexity that can be applied on a complex social network which contains static environments which are inspired from real life DBLP dataset. We used this network to solve team formation problem with WSCAN-TFP, GA, CA, HGA, and HCA.

3.1 Proposed Strategies to solve TFP

The five strategies we are using to solve the problem of TFP are listed below in section 3.1 and explained later in detail:

- Strategy 1 (S1)- WSCAN-TFP Weighted Structural Clustering Algorithm for Social Networks
- Strategy 2 (S2)- Genetic Algorithm (GA)
- Strategy 3 (S3)- Cultural Algorithm (CA)
- Strategy 4 (S4)- Hybrid Genetic Algorithm (HGA) with Schema
- Strategy 5 (S5)- Hybrid Cultural Algorithm (HCA) with Schema

3.2 Team Formation Problem (TFP): Definition

Problem 1. ***(Team Discovery)*** *Given a project P, a set of experts E, and a social network that is modeled as graph G, the problem of team discovery in social networks is to find a team of experts T for P from G so that the communication cost of T, defined as the frequent past collaboration of experts teamed up together. Then, the sum of distances of E is minimized. Weight W is the communication cost between experts E.*

3.3 Communication Cost

We want to find a team that satisfies a set of required skills with minimum communication cost. To measure communication cost, we are using direct and indirect connections. We want to form a team with least communication cost, so we prefer edge have least edge weight from one expert to another expert. To do so, we will check direct and indirect connections between the experts. With the hypothetical example in the figure 3.1 we can see how direct and indirect connections considered in order to minimize the distance or cost between experts.

$T =$ team
$E =$ set of "n" number of experts
$L =$ set of "m" skills
$P =$ project
$R =$ set of "k" requirement skills
whereas ; $R \subseteq L$

Figure 3.1: Team formation problem with the example

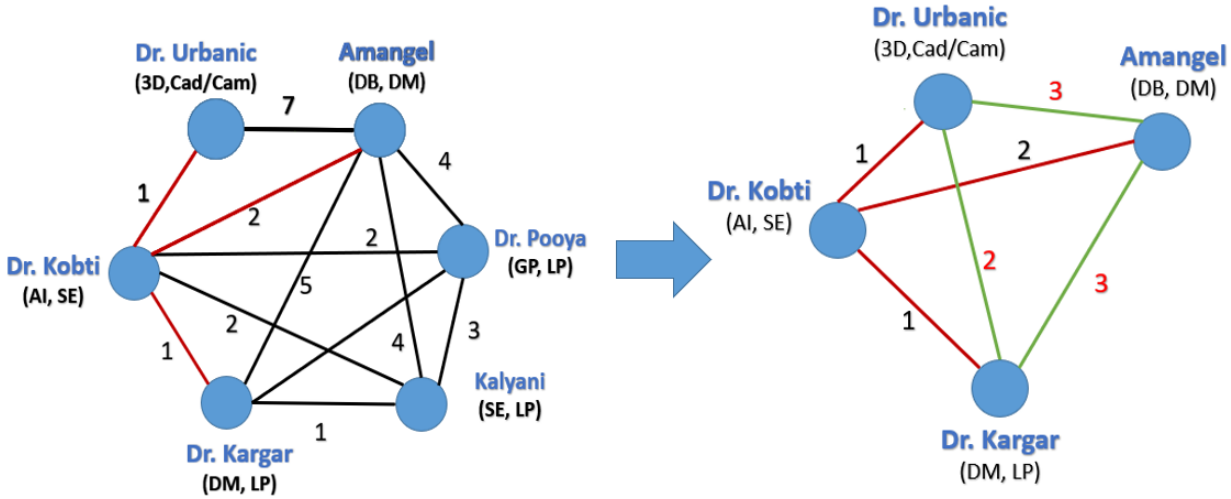


Figure 3.2: Direct and indirect connections in expert's network

To calculate communication cost we use sum of distance approach.

R = {DB, AI, 3D}
T = {Dr. Urbanic (3D), Dr. Kobti (AI), Amangel (DB)}

Team Communication Cost = (1+2+3) = 6

R = {DB, AI, 3D, LP}
T = {Dr. Urbanic (3D), Dr. Kobti (AI), Amangel (DB), Dr. Kargar (LP)}

Team Communication Cost = (1+2+3+2+1+3) = 12

Figure 3.3: Team (example)

We are using two methods to measure the performance of team based on communication cost.

3.3.1 Sum of distance function

Sum of distance can be defined as the summation of all the edge weights between team members.

Communication cost based on sum of distance function.

R = {DB, AI, 3D}

T = {Dr. Urbanic (3D), Dr. Kobti (AI), Amangel (DB)}

Team Communication Cost (sum of distance) = (1+2+3) = 6

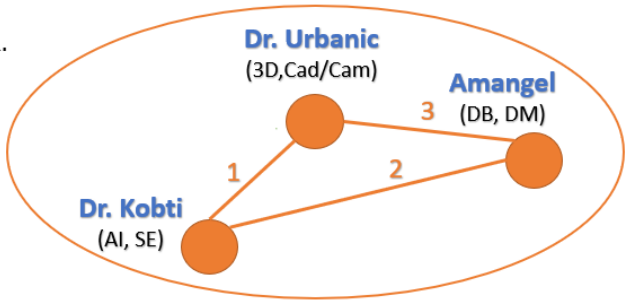


Figure 3.4: Communication cost based on sum of distance function (example)

3.3.2 Diameter function

The diameter function only measures the communication cost between the two experts that are furthest away from each other [26]

Communication cost based on diameter function.

R = {DB, AI, 3D}

T = {Dr. Urbanic (3D), Dr. Kobti (AI), Amangel (DB)}

Communication Cost (diameter) = 3 (maximum distance between two experts in a team)

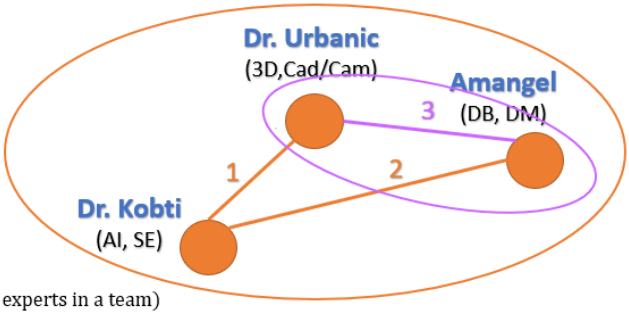


Figure 3.5: Communication cost based on diameter function (example)

3.4 Strategy 1 (S1) - WSCAN-TFP Weighted Structural Clustering Algorithm

We implemented WSCAN-TFP on social network graph to solve the Team formation problem within minimum communication cost or the sum of distances. The social network can be represented as a graph. In this thesis, we can use the network and graph interchangeably. A static, weighted graph G consists of a set of nodes V and a set of edges E. G = V, E). We represent the sizes of V and E as the set of experts

Symbol	Notation definition
E	Set of n experts
L	Set of m domain-specific skills
S	Set of skills of ith expert
n	Total number of experts
m	Total number of skills
k	Total number of required skills
e_i	ith expert
e_j	jth expert
l_i	ith skill
l_j	jth skill
R	set of project-specific required skills
R'	set of unfulfilled skills in project-specific required a skill set
G	a social graph of n nodes
D	relationship among experts or edge among E nodes
P	Project
T	Team of experts
$sumDistance$	Sum of distance
CC	Communication cost

Table 3.1: Notations used for S1

and a set of relationships between them. A graph may be directed or undirected for instance, a phone call may be from one party to another and will have a directed edge, or a mutual friendship may be represented as an undirected edge. Graphs may also be weighted, where there may be multiple edges occurring between two nodes (e.g., repeated text messages) or specific edge weights (e.g., monetary amounts for transactions, research paper co-authorship). In a weighted graph G, let $e_{(i,j)}$ be the edge between node i and node j. These nodes can be referred to as neighborhood nodes or incidental nodes on edge $e_{(i,j)}$. Graphs may be unipartite or multipartite. People in a group, papers in a citation network are examples of the unipartite social network. However, there are multiple classes of nodes and edges are only drawn between nodes of different classes, those social networks known as multipartite. Moreover, the social network can be represented as a graph either visually, or with an adjacency matrix A, where nodes are in rows and columns, and numbers in the matrix indicate the existence of edges. In the unweighted graph, the connection between nodes is represented as 0 or 1, whereas 1 indicates a link between two nodes and 0 is an indicator of no relationship between two nodes.

Assembling a team while considering optimization of communication cost will be an effective solution for TFP. The general problem is to assign the experts to a team T from a set of experts $e_{(i)}$possessing a set of skills $l_{(j)}$ to complete a project. However, to complete any project. We discover a team through a specific requirement criteria R.

We can create a graph representation of the Experts social network in the form $G(V, E, W)$, where

- Nodes V: represents the set of experts (E) represents
- Edges E: An edge $(i, j) \in D$ between two nodes $e_{(i)}$, $e_j \in E$ represents relationship among experts.

- Edge Weights W: The weight $w_{(i,j)}$ on an edge between nodes $e_{(i)}$, $e_j \in E$ is used to indicate the strength of the connection/relationship among experts (E).

3.4.1 Definitions related to Strategy 1 (S1)

Set of experts can be defined as a set of individuals $E; e_{(i)} \in E$ where $e_{(i)}, i = 1, 2, ..., n$ possess a set of skills and their prole represented by the skill space $L; l_{(j)} \in L$. Set of domain specific skills can be defined as a set of total number of abilities $l_{(j)}$, where $l_{(j)}, j = 1, 2, ..., m$ possessed by all experts available.

Set of project specific skills can be defined as a subset of abilities $l_{(j)}$, required to carry out a specific task with predefined criteria R, $[R]_{(k)} \subseteq l_{(j)} \in L, k = 1, 2, ..., x$. Project-specific skills, satisfying task requirement criteria to complete a task, is simply a subset of domain-specific skills set.

In this research, we focus on a social network modeled on weighted undirected graph G. An underlying social network connects the experts in E. Let $G = (E, D)$ be a graph with vertices (E) and edges (D) that are weighted W. Vertices indicate the set of experts and edges represent the previous collaboration between the connected experts. Terms such as node and expert can be used interchangeably in this work. As we have already discussed, we assume that individuals are organized in an undirected and weighted graph. Every node of G corresponds to an individual in $e_{(i)} \in E$. The edge weight W gives communication cost between two experts. If two experts have frequent collaborations, the edge weight is small and conversely, if the weight is large that means rare collaborations occurred. For example, if two experts work on many projects in their past experience, their strength of connectivity is high, it means the distance between them is low.

Suppose, each expert $e_{(i)}$ has a set of skills $S_{(e_{(i)})} \subseteq L$. To be part of a team or to be member of a task or project team, every expert must have at least one skill from R, $R \subseteq l_{(j)}$ and $[l]_{(j)} \in L$. Therefore, if at least one element of R is satisfied by any $E; e_{(i)} \in E, i = 1, 2, ...n$ from set of n experts. Then, she/he is a member of the team.

$E = e_{(1)}, e_{(2)}, ..., e_{(n)}$ species a set of n experts, and $L = [l]_1, [l]_2, ..., [l]_m$ species a set of m skills. Each experte $e_{(i)}$ has a set of skills, specied as $S_{(e_{(i)})}$ and $S(e_{(i)}) \subseteq L$. If $[l]_{(j)} \in S_{(e_{(i)})}$, expert $e_{(i)}$ posses skill $[l]_{(j)}$. A subset of experts $E' \subseteq E$ have skill $[l]_{(j)}$ if at least one of them posses $l_{(j)}$. For each skill $[l]_{(j)}$, the set of all experts that posses skill $[l]_{(j)}$ is specied as $E([l]_{(j)}) = e_{(i)}|[l]_{(j)} \in S_{(e_{(i)})}$. A project $P = [l]_1, [l]_2, ..., [l]_t$ is composed of a set of R skills that are required to be completed by some experts.

Definition 13. ***(Team of Experts)*** *Given a set of experts E and a project P that needs a set of skills $\{el_1, el_2, \ldots, el_m\}$, a* team T of experts *for P is a set of R skill-expert pairs:*

$T = \{\langle e_{l_{j=1}}\rangle, \langle e_{l_{j=2}}\rangle, \ldots, \langle e_{l_{j=r}}\rangle\}$,

where e_{l_j} is an expert that posses skill l_j for $j = 1, \ldots, m$. This means expert e_{l_j} is responsible for skill l_j.

Definition 14. ***(Sum of Distances)*** *Given a graph G and a team of experts $\{ T = \langle e_{l_1}\rangle, \langle e_{l_2}\rangle, \ldots, \langle, e_{l_r}\rangle\}$, the* sum of distances *of the team is defined as sumDistance. where $dist(e_{l_i}, e_{l_j})$ is the distance between e_{l_i} and e_{l_j} in G (i.e., the sum of weights on the path between e_{l_i} and e_{l_j}).*

$$sumDistance = \sum_{i=1}^{x} \sum_{j=i+1}^{y} dist(e_{l_i}, e_{l_j}) \tag{3.1}$$

Symbol	Notation definition
$\tau(e_i)$	vertex structure of e_i node or expert
$Core$	Core expert or node
ϵ	distance from core node
N_ϵ	neighbourhood of ith expert/node with ϵ
σ	structural similarity
μ	Total number of neighbours connected to a node
w	weight between two nodes or experts
S_{e_i}	Set of skill for ith expert
E	Set of experts
e_i	ith expert
e_j	jth expert

Table 3.2: WSCAN notations used for S1

Definition 15. ***(Communication Cost (CC))*** *can be defined a distance between two experts $e_{(i)}$ and $e_{(j)}$ on a graph G. In this paper, $CC(e_{(i)}, e_{(j)})$ and edge weight $(e_{(i)}, e_{(j)})$ are used interchangeably.*

3.4.2 WSCAN Definitions

We are using weighted structural clustering algorithm (WSCAN) on social network as a graph to find the best team of experts.

Definition 16. ***(Vertex structure)*** *Let $e_i \in E$, the structure of e_i is defined by its neighborhood, denoted by $\tau(e_i)$.*

$$\tau(e_i) = \{e_j \in E \vee (e_i, e_j) \in E\} \cup \{e_j\} \tag{3.2}$$

Definition 17. ***(ϵ - Neighborhood)***

$$N_\epsilon = \{e_j \in \tau(e_i) | \sigma(e_i, e_j) \geq \epsilon\} \tag{3.3}$$

Definition 18. ***(Extended Structural Similarity)*** *Structural similarity of two vertices, suppose e_i (ith expert) and e_j (jth expert) will be large if they share a similar structure of neighbors that are the frequent regime of working together and communication cost.*

$$\sigma(e_i, e_j) = \frac{|\tau(e_i) \cap \tau(e_j)|}{\sqrt{|\tau(e_i||\tau(e_j|)}} w(e_i, e_j) \tag{3.4}$$

Where w is weight of the edge connecting two vertices e_i and e_j. If σ is inversely proportional to communication cost. If σ is high, communication costCC will be low.

$$\sigma(e_i, e_j) \propto \frac{1}{CC(e_i, e_j)} \tag{3.5}$$

Relationship of communication cost CC and strong/weak bonding f between experts. The weight of communication cost CC_e of experts is inversely proportional to the frequent regime of collaboration f_e of experts E.

$$CC(e_i, e_j) \propto \frac{1}{f(e_i, e_j)} \tag{3.6}$$

Suppose, if communication is more in e_i and e_j that means e_i and e_j works less frequently with each other and vice versa. Thereof, less communication cost represents the strong bond between e_i and e_j and more cost shows weak bonds between e_i and e_j. Therefore, Strong bonds between e_i and e_j gives high structural similarity. σ is directly proportional to $f(e_i.e_j)$.

$$\sigma(e_i, e_j) \propto f(e_i, e_j) \tag{3.7}$$

If two experts e_i and e_j collaborate together more frequently they are likely to have more structural similarity.

Definition 19. ***(Communication cost)*** *Communication cost CC can be defined a distance between two experts e_i and e_j on a graph G. On large networks such as 50k, 100k and 200k its time consuming. So, to overcome this limitation we have used 2 hop cover as discussed in [16], [3]. In this paper, communication cost $CC_{(e_i,e_j)}$ and edge weight $w_{(e_i,e_j)}$ are used interchangeably.*

Definition 20. ***(Core Expert)*** *Let $\epsilon \in \Re$ and $\mu \in N$. A vertex $e_i \in E$ is called a core with reference to ϵ and μ , if its ϵ - neighborhood contains at least μ vertices.*

$$Core_{\epsilon,\mu}(e_i) \leftrightarrow |N_\epsilon| \geq \mu \tag{3.8}$$

Where μ is the number of neighborhood experts connected to core vertex (highly connected expert) with minimum distance or minimum communication cost.

Definition 21. ***(Project or Task)*** *Project or task P can be defined as a piece of work to be completed by an eligible team T with the expertise on the set of project specific skills R.*

Definition 22. ***(collective expertise)*** *It can be defined as a phenomenon of occurrence of the certain level of expertise among the group of individuals or team members*

in a team T who are possessing set of project specific skills R necessary to complete a task or project P as a team with minimum cost CC.

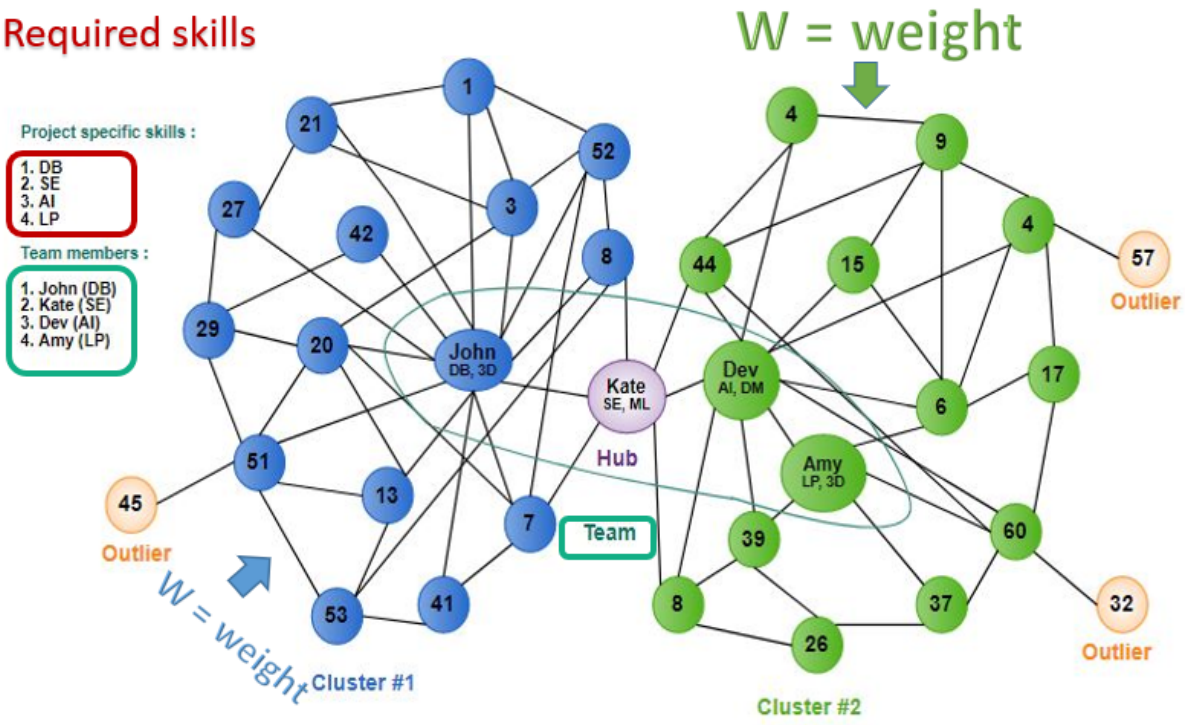


Figure 3.6: WSCAN-TFP (example)

We implemented the same version of the WSCAN-TFP algorithm in [41]on G1 graph and now with other combinations to compare experimental results from [41]. In [41], We generated the 50K nodes of equal edge weight graph with 1.0 of weight on all edges. Now, we conducted more experiments with this approach on 50K and 100K nodes network.

The SCAN requires threshold value to form structural similarity with neighborhood nodes [41]. Therefore, we tested with the different number of skills to find them at most value as a threshold [41]. The experiment has been shown with the communication cost vs a number of required skill graph.

The experiment always begins by calculating communication cost from Core expert to the neighborhood [41]. Therefore, we calculated the value of communication cost of the team of experts with required skills [41]. The figure below shows the comparison of the communication cost of a team for the set of the required number of skills.

Input: Graph $G = (< E, D >, \epsilon, \mu), W(e_i, e_j)$; $W(e_i, e_j) = CC(e_i, e_j)$ input project P has a list of R project specific skills required to complete given project, We have domain specific skills L of each expert E. $\{l_1, l_2, \ldots, l_m\}$; set of domain specific skills of each expert e_i, $S_{(e_i)}$, R is set of project specific skills, where $R \subseteq L$
Output: Best Team T

Initializations
$E \in 1 \geq i, j \leq n \leftarrow$ number of n experts
$L \in 1 \geq i, j \leq m \leftarrow$ number of m domain specific skills
$R \in 1 \geq i, j \leq x \leftarrow$ number of x required project skills
$PoE \in 1 \geq i, j \leq z \leftarrow$ number of experts have atleast one skill from R
Begin
find Pool of experts (PoE) from E
All vertices are unclassified;
for
each unclassified vertex $e \in E$
// Step 1 : Check if e is core;
if $Core_{\epsilon,\mu}(e)$ **then**
// Step 2.1 : if e is a Core, a new cluster is formed;
generate new clusterID;
insert $e_i \in N_\epsilon(e)$ into queue Q;
while $Q \neq 0$ **do** e_j = first expert in Q
if e_i is unclassified or non-member **then**
assign current clusterID to e_i;
if e_i is unclassified **then**
insert e_i into queue Q;
remove e_j from Q;
else
// Step 2.2 if e is not core, it is labeled as non member label of e as non-member;
for ends here.
// Step 3 : Search project specific skills R in classified members.
for
every expert $e \in E$, calculate distance from Core vertex e_i **then**
// Step 3.1 : from $Se_i \subseteq L$; check if e_i have project specific skill
$R_k \in R$; $L \supseteq R$
if e_i and e_j same skill as mentioned in R
if else choose based on less communication cost
else choose randomly out of above two
remove skill already found from list = $ReqS$
// Step 3.2 : **if** e_i have any $L \supseteq R_{i,j}$; **then**
put e_i in T

36: **while** team
37: $T \neq 0$ **do** first is e_i member of the team T.
38: remove requirement of $Core_{\epsilon}, \mu(er_i, j)$ skill any further; **else**
39: // Step 3.3 : increase communication cost CC
40: repeat steps 3.1, 3.2 and 3.3
41: **if** $e_{i,j}$ is common to cluster 1 and cluster 2
42: label it as a hub.
43: **else** check for communication cost
44: if CC is highest and weak ties with other experts
45: label it as a outlier
46: check for required skills,
47: **while** checking for hubs, **do** try for possible minimum cost,**if** yes,put in team list T **else** search further by increasing CC,search till $T = ReqTEAMmembers$ $E' \subseteq E$
48: **for** ends here
49: return T
50: ends WSCAN

Algorithm 1: WSCAN-TFP

3.4.3 Proposed Solution/algorithm with Strategy 1 (S1)

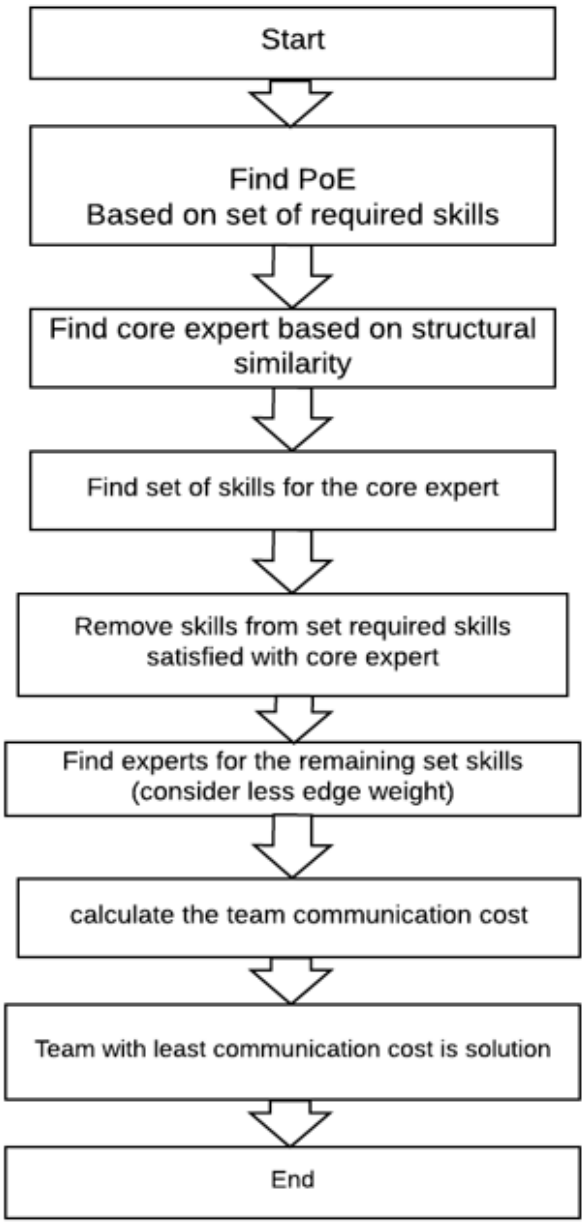


Figure 3.7: WSCAN-TFP (flowchart)

3.5 Strategy 2 (S2) - Genetic Algorithm (GA)

Genetic Algorithms begin with a randomly generated population at time zero. Each iteration of the time counter yields a new generation. During any generation, the population is referred to as the search space [17]. External pressure is modeled by a fitness function that assigns (positive) numerical values to candidate solutions. A random process called selection determines which solutions survive to the next generation, but solutions with low fitness values have a lower probability of survival [17]. Reproduction is mimicked via operations by which existing solutions produce new solutions [17]. Crossover operator and the mutation operator helps the solution to evolve until termination condition that is predetermined, the algorithm halts and returns the best solution.

Social network graph generated with Experts possessing the set of skills. After generating the graph with weights calculated based on methods mentioned in section 4.2. We start to filter out those experts who has at least one skill from the set of required skills.

To find the best team with experts who are possessing at least one skill from the set of required skill is our objective of using this strategy. In strategy (S2), we use the genetic algorithm (GA), which is a search heuristic method and used to solve optimization problems. We want to search for a set of experts with the set of required skills in minimum possible communication cost.Communication cost is our parameter we keep under consideration while solving this problem. We desire to keep its value as minimum as possible with all the required skills needed to complete a task.

In (S2), first, we initialize the random population that contains a set of individuals or chromosomes. Each individual is a potential solution to the problem. However,

For every solution, it's fitness is measured by the fitness function. In this thesis, we consider communication cost or the sum of the distance between experts as a fitness function to calculate the fitness of potential teams or each individual solution.

Moreover, to find the fittest individual we use selection, crossover and mutation procedure until the termination condition meets.For evaluation of Genetic algorithm to solve team formation problem, We implemented this algorithm using social network graph in chapter 4.

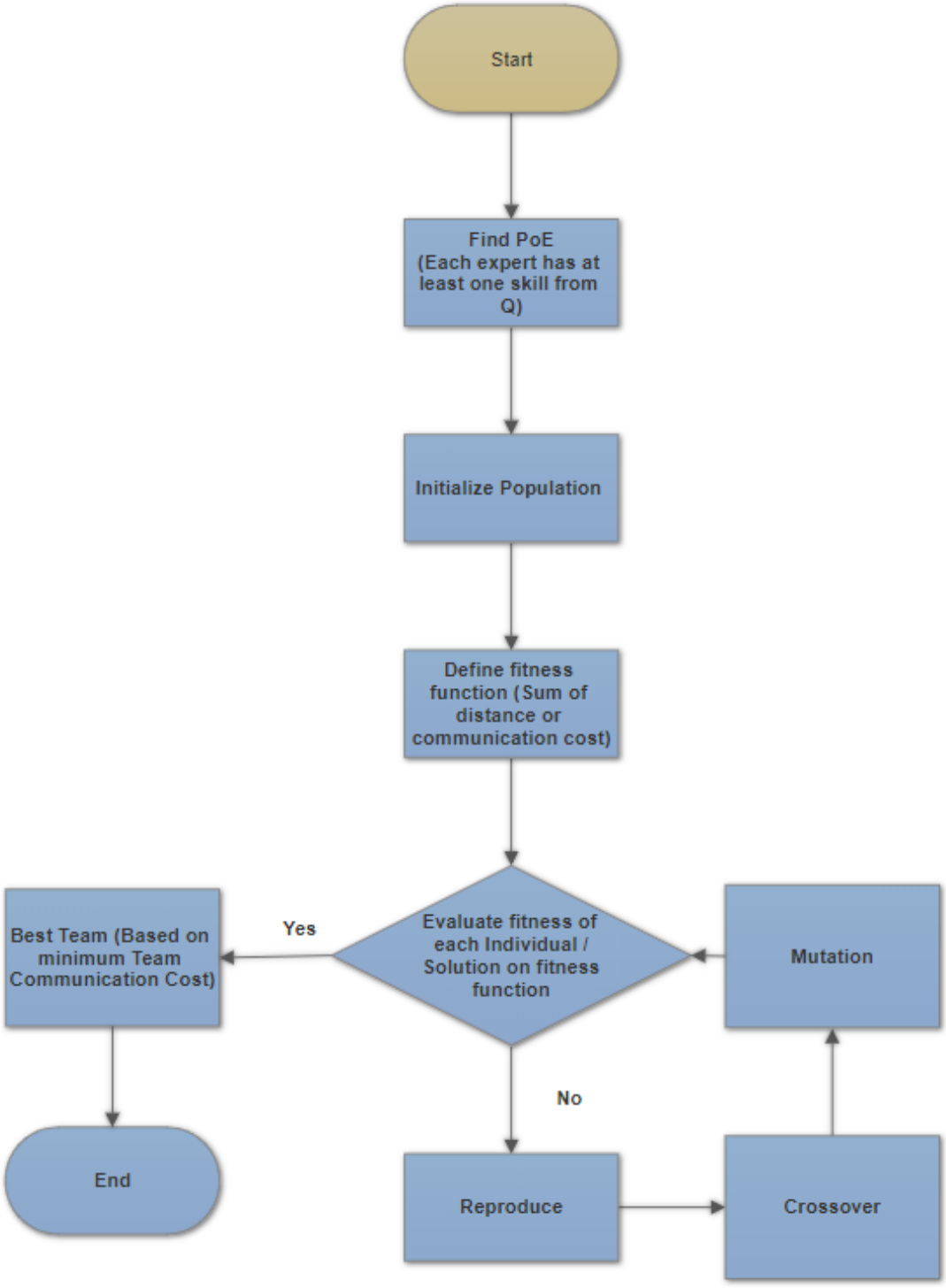


Figure 3.8: Flowchart showing Genetic Algorithm (GA) for Team Formation Problem

3.6 Strategy 3 (S3) - Cultural Algorithm (CA)

Cultural Algorithms are computational self-adaptive models which consist of two main components that are population and a belief space. Problem-solving experience of individuals selected from the population space by the fitness function is generalized and stored in the belief space. This knowledge can then controlled and utilized for the evolution of the population component [15].

The cultural algorithm has a similar workflow to the genetic algorithm with an additional component that is known as a belief space. First, we generate a random population similar to in genetic algorithm. Later, the fitness score is measured based on fitness function similar to (S2). However, crossover and mutation procedure is used to produce better offspring with every iteration.

For evaluation of Cultural algorithm to solve team formation problem, We implemented this algorithm using social network graph. Detailed experimental results are shown in chapter 4.

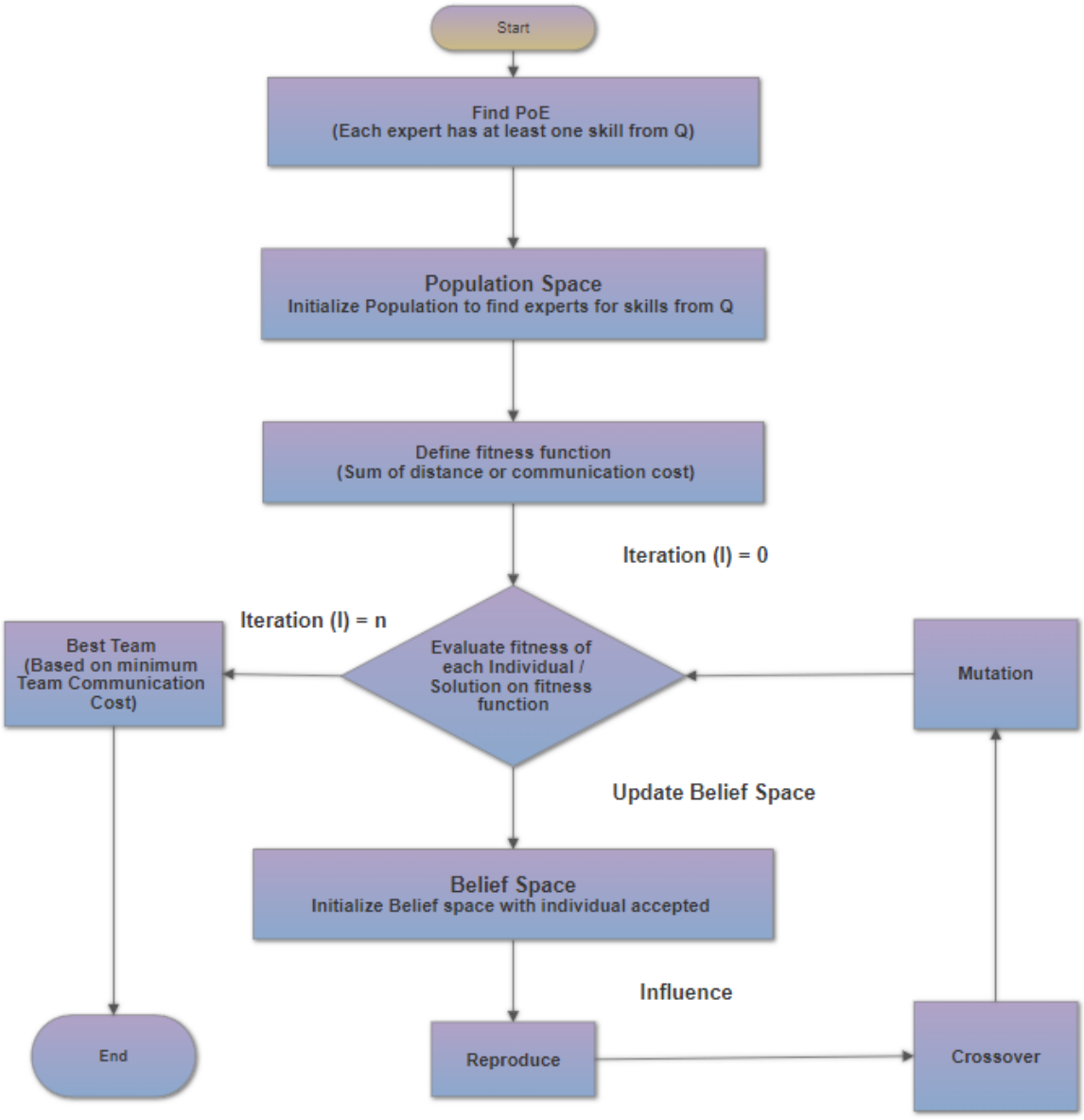


Figure 3.9: Flowchart showing S3 for Team Formation Problem

3.7 Strategy 4 (S4) - Hybrid Genetic Algorithm (HGA) using Schema

3.7.1 Definitions related to Strategy 4 (S4)

In this section, we present the problem with various definitions for TFP. Therefore, assembling a team while optimizing the communication cost will be an effective solution to this problem.

The general problem is to allocate experts in a team T from a set of experts p_i possessing a set of skills l_j to complete a task or project. However, to complete this project, we find a team through specific requirement criteria Q.

In this section, we describe in detail the general framework.

Set of experts can be written as a set of individuals P ; $p_i \in P$, where $p_i = \{p; i = 1, 2, \ldots, n\}$ have a set of skills and their profile is represented by the skill space L ; $\{l_j \in L\}$.

Set of total domain specific skills can be written as a set of total number of abilities l_j, where $l_j = \{l; j = 1, 2, \ldots, m\}$ held by all available experts .

Set of project specific skills can be written as a subset of abilities l_j, required to carry out a specific project with predefined given criteria Q.

$\{ Q_k \subseteq l_j \in L\}$ $\{Q; k = 1, 2, \ldots, x\}$

Skills or abilities satisfying project specific requirement criteria to complete a task is simply a subset of total domain-specific skills set.

Symbol	Notation definition
P	Set of n experts
L	Set of m domain specific skills
S	Set of skills of ith expert
n	Total number of experts
m	Total number of skills
k	Total number of required skills
p_i	ith expert
p_j	jth expert
$S(p_i)$	Set of skills
l_i	ith skill
l_j	jth skill
Q	set of project specific required skills
Q'	set of unfulfilled skills in project specific required skill set
G	a social graph of n nodes
C	relationship among experts or edge among P nodes
$Proj$	Project
T	Team of experts
$dist(p_i, p_j)$	Sum of distance between two or more experts
CC	Communication cost
H	Schema template
SS	Serach space
$o(H)$	Order of schema
$f(H, t)$	Average fitness
$Core$	Core expert
I	Number of iterations
I_{max}	Maximum number of iterations

Table 3.3: Notations used for S4 and S5

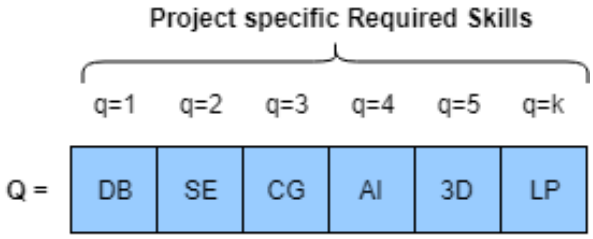


Figure 3.10: Required skills necessary to complete a Project

Team for task If skill $\{Q \in p_i\}$ then we can say that $\{p_i \in T\}$ and it is satisfying the requirement of the team T. We can also write the cover of a set of individuals P with respect to team for project J, denoted by $cov\{P', J\}$, to be the set of skills that are required by T and for which there exists at least one individual in P that has them.

$P(l_j) = \{p_i | l_j \in Q(p_i)\}$.

We focused on a social network modeled on the weighted undirected graph G. An underlying social network connects the experts in P. Let $G = (P, C)$ be a graph with vertices (P) and edges (C) and edges are weighted W. Whereas, Vertices or nodes indicate a set of experts (set of vertices) and edges show a set of pairs that give frequent regime of working together or collaboration previously between two experts for a specific project. Terms such as a node and experts, defined bits or fixed gene, undefined bits or * bits can be used interchangeably in this work.

As we have already discussed, we assume that individuals are organized in an undirected, but the weighted graph. Every node in G corresponding to an individual in p_i ; { $p_i \in P$} and C is the set of edges connecting the nodes. Where W are the edge weight that gives us the distance between two experts. The distance between two experts p_i and p_j, specified as $dist(p_i, p_j)$, is equal to the sum of the weights on the shortest path between them in the input graph G.

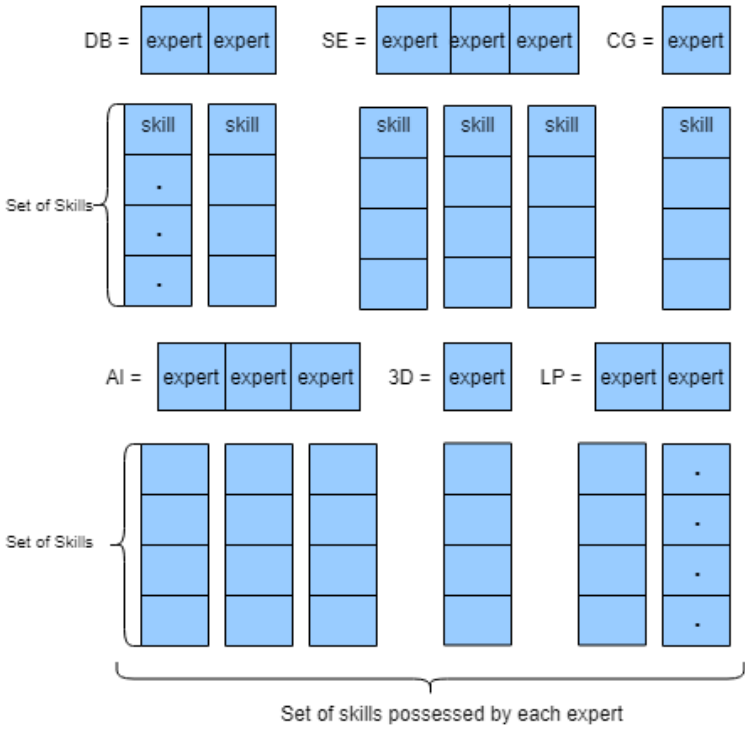


Figure 3.11: Expert list for Required skills necessary to complete a Project. Every Expert has a Set of skills

Assume, each expert P_i has a set of skills $S(p_i) \subseteq L$. To be member of the team or to be part of project team, every expert must have at least one skill from Q whereas $\{\ Q \subseteq l_j\}$ and $\{l_j \in L\}$ Thereof, if at least one element of Q (i.e. at least one skill from set of project skills) is satisfied by any P ;$\{\ p_i \in P\}$ $\{i = 1, 2, \ldots n\}$ from set of n experts. Therefor, expert is final member of team.

$P = \{p_{i=1}, p_{i=2}, \ldots, p_{i=n}\}$ specifies a set of n experts, and $L = \{l_{j=1}, l_{j=2}, \ldots, l_{j=m}\}$ specifies a set of m skills. Each expert p_i has a set of skills, specified as $S(p_i)$, and $S(p_i) \subseteq L$. If $l_j \in S(p_i)$, expert p_i posses skill l_j. A subset of experts $P' \subseteq P$ have skill l_j if at least one of them posses l_j. For each skill l_j, the set of all experts that posses skill l_j is specified as $P(l_j) = \{p_i | l_j \in S(p_i)\}$. A project $J = \{l_1, l_2, \ldots, l_t\}$ is composed of a set of Q skills that are required to be completed by some experts. A subset

of experts $P' \subseteq P$ is able to *complete* a project J if $\forall Q_k \in J \; \exists \; p_i \in P', l_j \in S(p_i)$.

Definition 23. ***(Team of Experts)*** *can be defined in a given a set of experts P and a project J that needs a set of skills $\{pl_1, pl_2, \ldots, pl_n\}$, a* team T of experts *for J is a set of Q skill-expert pairs.*
where p_{l_j} is an expert that posses ability l_j for $j = 1, \ldots, m$. This means expert p_{l_j} is responsible for skill l_j.

$$T = \{\langle p_{l_{j=1}} \rangle, \langle p_{l_{j=2}} \rangle, \ldots, \langle p_{l_{j=r}} \rangle\} \tag{3.9}$$

Definition 24. ***(Sum of Distances)*** *Given a graph G and a team of experts $\{ \; T = \langle p_{l_1} \rangle, \langle p_{l_2} \rangle, \ldots, \langle , p_{l_r} \rangle\}$, the* sum of distances *of the team is defined as sumDistance.where $dist(p_{l_i}, p_{l_j})$ is the distance between p_{l_i} and p_{l_j} in G (i.e., the sum of weights on the path between p_{l_i} and p_{l_j}).*

$$sumDistance = \sum_{i=1}^{x} \sum_{j=i+1}^{y} dist(p_{l_i}, p_{l_j}) \tag{3.10}$$

Problem 2. ***(Finding a team)*** *Given a project J, a set of experts P, and a social network that is modeled into a graph G, the problem of team search in social networks is to find a team of experts T for J from G so that the communication cost of T, defined as the past collaborations of experts joined up together. However, the sum of distances of P is minimized. Weight W is the communication cost between experts P.*

3.7.2 Proposed Algorithm for Strategy 4 (S4)

We are using weighted structural clustering algorithm (WSCAN) on social network as a graph to find the core expert in a network to find best team.

Definition 25. ***(Vertex structure)*** *Let $p_i \in P$, the structure of p_i is defined by its neighborhood, denoted by $\tau(p_i)$.*

$$\tau(p_i) = \{p_j \in P \vee (p_i, p_j) \in P\} \cup \{p_j\} \tag{3.11}$$

Definition 26. ***(ϵ - Neighborhood)***

$$N_\epsilon = \{p_j \in \tau(p_i) | \sigma(p_i, p_j) \geq \epsilon\} \tag{3.12}$$

Definition 27. ***(Extended Structural Similarity)*** *Structural similarity of two nodes, suppose p_i (ith expert) and p_j (jth expert) will be large if they share a similar structure of neighbors that is frequent regime of working together and communication cost.*

Where w is weight of the edge connecting two nodes p_i and p_j. If σ is inversely proportional to communication cost. If σ is high, communication costcc will be low.

$$\sigma(p_i, p_j) = \frac{|\tau(P_i) \cap \tau(p_j)|}{\sqrt{|\tau(p_i||\tau(p_j|)}} w(p_i, p_j) \tag{3.13}$$

$$\sigma(p_i, p_j) \propto \frac{1}{cc(p_i, p_j)} \tag{3.14}$$

Relationship of communication cost cc and strong/weak bonding f between experts. The weight of communication cost cc_p of experts is inversely proportional to the frequent regime of collaboration f_p of experts P.

$$cc(p_i, p_j) \propto \frac{1}{f(p_i, p_j)} \tag{3.15}$$

Suppose, if communication is more in p_i and p_j that means p_i and p_j works less frequently with each other and vice versa. Thereof, less communication cost represents strong bond between p_i and p_j and more cost shows weak bonds between p_i and p_j. Therefore, Strong bonds between p_i and p_j gives high structural similarity. σ is directly proportional to $f(p_i.p_j)$.

$$\sigma(p_i, p_j) \propto f(p_i, p_j) \tag{3.16}$$

If two experts p_i and p_j collaborates together more frequently they are likely to have more structural similarity.

Hybrid Genetic Algorithms (HGA) we are proposing a novel approach which can be divided into three main parts. In First part, we are using a modified version of WSCAN that is a non-knowledge based algorithm used for find out the highly connected expert in the graph based on structural similarity. This core expert is not randomly select; rather we follow the steps used in the original SCAN approach proposed by [51]. But this simple SCAN was limited to small graphs. To overcome this limitation SCAN++ was proposed by [43]. However, to utilize structural similarity

Input: Graph $G = (< P, C >, \epsilon, \mu), W(p_i, p_j)$; $W(p_i, p_j) = cc(p_i, p_j)$ input project P has a list of Q project specific skills required to complete given project, We have domain specific skills L of each expert E. $\{l_1, l_2, \ldots, l_m\}$; set of domain specific skills of each expert p_i, $S(p_i)$, Q is set of project specific skills, where $Q \subseteq L$
Output: Best Team T

Initializations
$P \in 1 \geq i, j \leq n \leftarrow$ number of n experts
$L \in 1 \geq i, j \leq m \leftarrow$ number of m domain specific skills
$Q \in 1 \geq i, j \leq x \leftarrow$ number of x required project skills
$PoE \in 1 \geq i, j \leq z \leftarrow$ number of experts have atleast one skill from R
Begin
find and store Pool of experts (PoE) from P
All vertices are unclassified;
for
each unclassified vertex $p \in P$ **do**
// Step 1 : Check if p is core $Core_{\epsilon,\mu}(p)$ **then**
generate new clusterID;
insert $p_i \in N_\epsilon(p)$ into queue U; **ends if**
while $U \neq 0$ **do** p_j = first expert in U
if p_i is unclassified or non-member **then**
assign current clusterID to p_i;
if p_i is unclassified **then**
insert p_i into queue U;
remove p_j from U;
else
// Step 2 if p is not core, it is labeled as non member label of p as non-member;
if ends here
while ends here
for ends here
ends WSCAN

Algorithm 2: modified WSCAN-TFP

1: **Phase 1- Begin**
2: find PoE expert (WSCAN-TFP)
3: find Core expert (WSCAN-TFP)
4: **Phase1- End**
5: **Phase2- Begin**
6: Search project specific skills R for Core
7: Assign bit string in schema H
8: Fix Core $[Core]_{p(i,j)}, *, *, *$ in H
9: fill * string bits from $Q_k \in Q$; $L \supseteq Q$
10: **if** p_i and p_j same skill as mentioned in Q
11: **else** choose randomly out of above two
12: $[Core]_{p(i,j)}, p_{(i,j)} \in H$
13: **Phase2- End**
14: **Phase3- Begin**
15: **initialize the population** with H
16: current generation = 1
17: **for** $p(i,j) = 1$ to t **do**
18: evaluate a population
19: for $p_(i,j) = 1$ to t **do**
20: Evaluate $T(p_(i,j))$ on $f(H,t)$
21: **Selection**
22: **Crossover** for $p_(i,j) = 1$ to t **do**
23: if $(j <= crossoverpoint)$
24: offspring $[i][j]$ = parents $[i][j]$
25: offspring $[i+1][j]$ = parents $[i+1][j]$
26: **else**
27: offspring $[i][j]$ = parents $[i+1][j]$
28: offspring $[i+1][j]$ = parents $[i][j]$
29: **mutation**
30: for $p_(i,j) = 1$ to t do
31: if random number ¡ mutation rate
32: mutation $(offspring[p_i])$
33: evaluate the offspring
34: **new population**
35: $popl$ = offspring
36: current generation + = t
37: evaluate on $f(H,t)$ at generation t
38: **termination condition**
39: **best team** $T([Core]_p(i,j), p_(i,j))$
40: ends HGA
41: **Phase3- Ends**

Algorithm 3: Hybrid Genetic Algorithm(HGA) with Schema

based clustering concept, we considered WSCAN approach published in [14] and in [13].Later, to utilize this structural based clustering concept for team formation problem, a modified version of WSCAN namely, WSCAN-TFP algorithm was published in [41] research paper. Moreover, after selecting the core expert, we are constructing a schema. This construction of schema is inspired from schema theory proposed by John Holland.

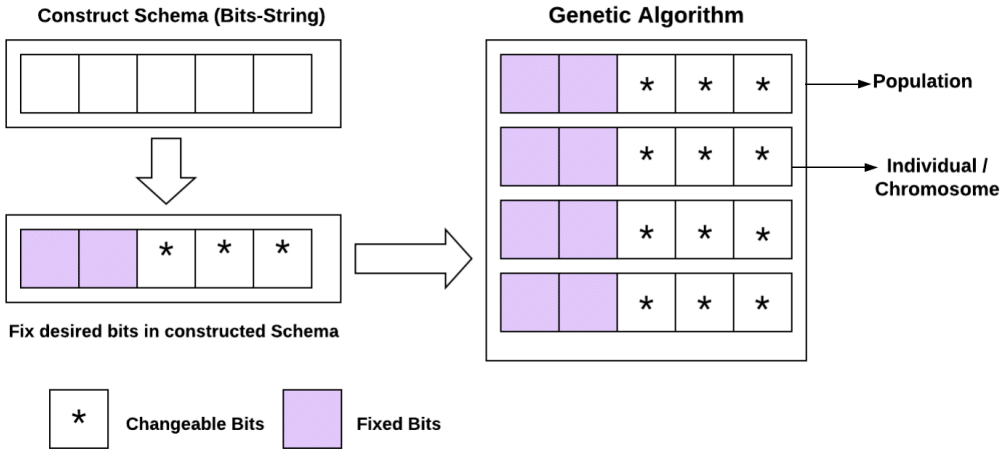


Figure 3.12: Framework of Hybrid Genetic Algorithm with Schema.

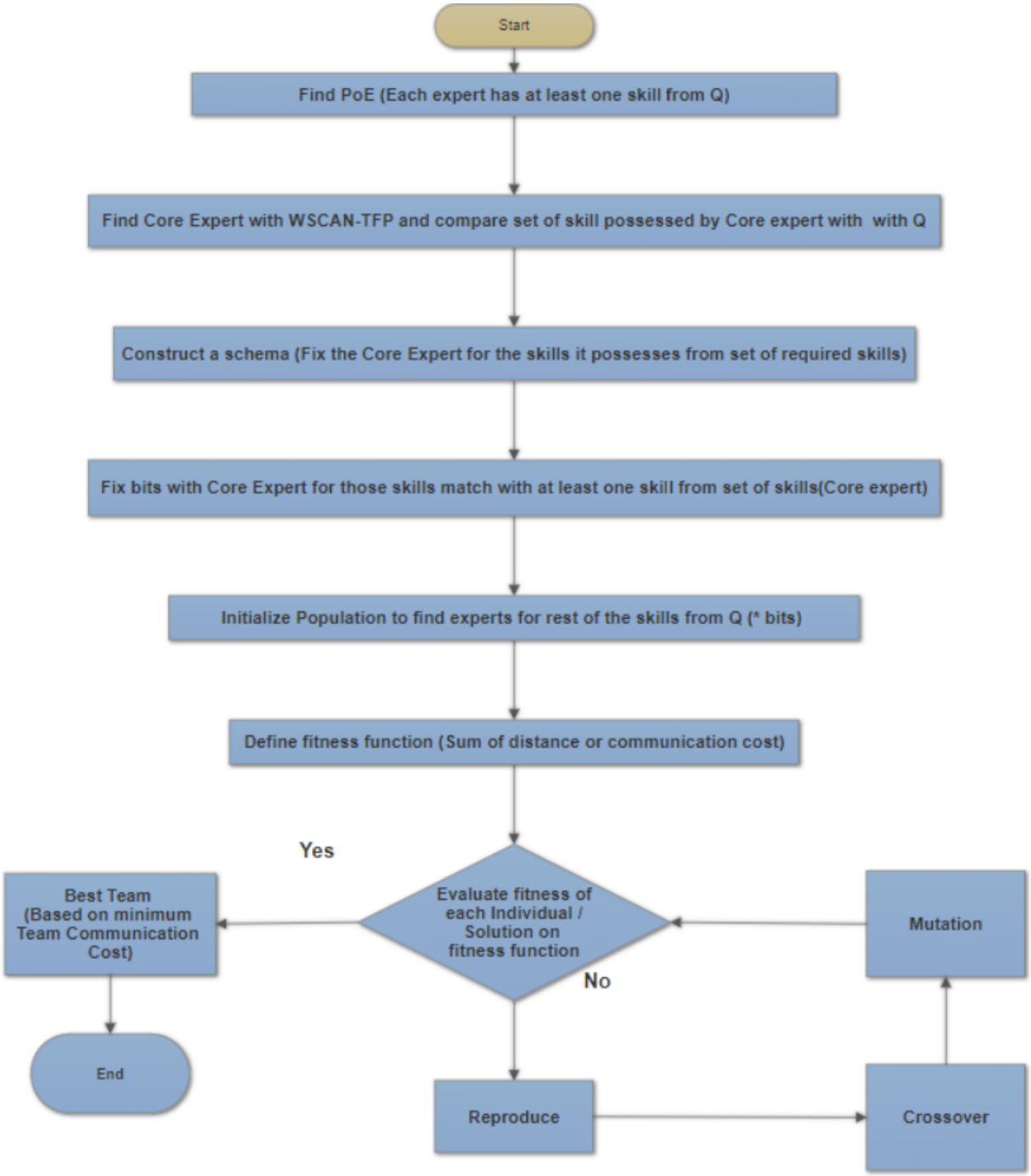


Figure 3.13: Flowchart of Hybrid Genetic Algorithm with Schema for TFP.

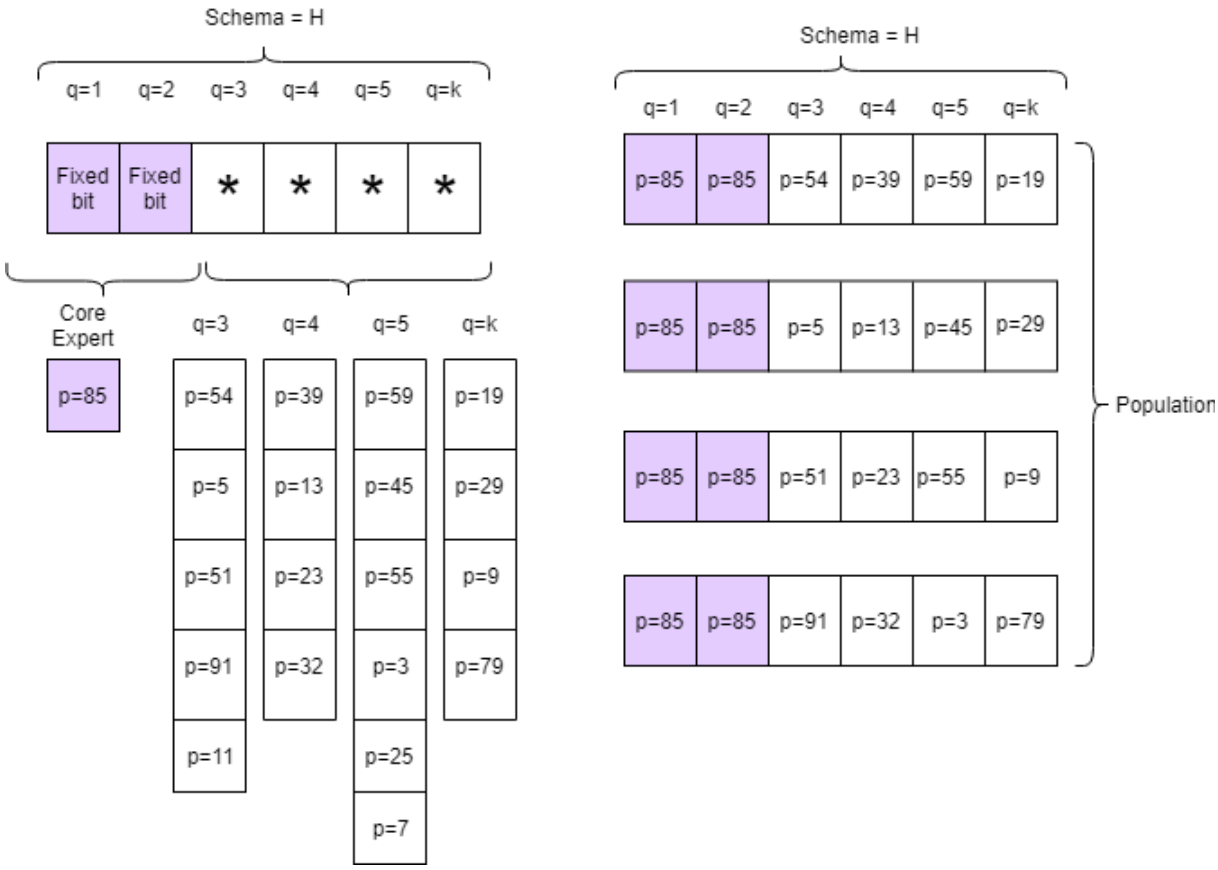


Figure 3.14: Working of schema template based approach with hybrid genetic algorithm. Where * shows changeable bits over the process of crossover and mutation

3.7.3 Schema Theorem - Definitions

Schema theorem is used to increase the possibility of desired results and minimize the likelihood of undesired results for a given problem. We construct schema template for a given problem to find a best solution and construction of this schema template helps in achievement of desired results. Few definitions are explained below; these definitions are based on schema theorem by John Holland [25], [24].

Definition 28. ***(Schema** H**)*** *A schema is a subset of the search space ($H \subset SS$) of all possible individual solutions for which all the genes match the template for schema H.*

Definition 29. ***(Order of Schema*** $o(H)$ ***)*** *The order of schema* $o(H)$ *is number of defined bits(non-asterisk) in schema* H*.For example* $o(1**0*) = 2$

Definition 30. ***(Instance of Schema*** (H) ***)*** *A bit string that belongs to a schema called an instance of a schema. It can be denoted by* A*, which shows a number of asterisks in a schema and instance of the schema is* 2^A*.*

Definition 31. ***(Number of Instances of Schema*** (H) ***)*** *Notation* $m(h,t)$ *shows number of instances in a schema* H *at* t^{th} *generation in a genetic algorithm.*

Definition 32. ***(Fitness of population at generation*** t ***)*** $f(x)$ *shows fitness of bit string* x *and* $f(H,t)$ *denotes average fitness of instances of the schema in the population at* t^{th} *generation.*

$$f(H,t) = \frac{\sum_{x \in H} f(x)}{m(H,t)} \tag{3.17}$$

Definition 33. ***(Schema Template)*** *Schema template is made up of bits in a string or genes in an individual. In this paper, we are fixing core expert and masking it to change it in the crossover and mutation process. Where * are the changeable bits during the process to find an efficient team over the process.*

$$H = ([Core]_{p(i,j)}, *, *, *) \tag{3.18}$$

Definition 34. ***(Core Expert)*** *Let $\epsilon \in \Re$ and $\mu \in N$. A vertex $p_i \in P$ is called a core with reference to ϵ and μ , if its ϵ - neighborhood contains at least μ vertices.*

$$Core_{\epsilon,\mu}(p_i) \leftrightarrow |N_\epsilon| \geq \mu \tag{3.19}$$

Where μ is the number of neighborhood experts connected to core vertex (highly connected expert) with minimum distance or minimum communication cost.

Definition 35. ***(Project or Task)*** *Project or task J can be defined as a piece of work to be completed by an eligible team T with the expertise on the set of project specific skills Q.*

Strategy 4 follows the procedure of the hybrid genetic algorithm (HGA) after generating a social network graph. Results are given in chapter 4 in detail.

3.8 Strategy 5 (S5) - Hybrid Cultural Algorithm (HCA)

In this approach, we are solving the same problem defined earlier with minimum communication cost. This method is different from the previous solution because in expert selection strategy (S1), we are using non-knowledge based modified clustering algorithm using structural similarity and threshold. Whereas, Expert selection strategy (S4) utilizes the advantage of knowledge-based hybrid algorithm model with

non-knowledge based modified algorithm (S1) concept of structural similarity, clustering, and threshold.

However, in S5, we have three main parts to explain. In First part, we are using WSCAN-TFP for social network graphs to find a Core expert. In the second part of the Algorithm, we are constructing a schema. Schema construction helps us to achieve our desired results. In the schema, we assign the bit to a string. we fix some bits, which will not change throw out the process and some bits are shown as * will be changed in population space. * bits are different in each individual / solution, whereas fixed bits will remain same throughout the process.

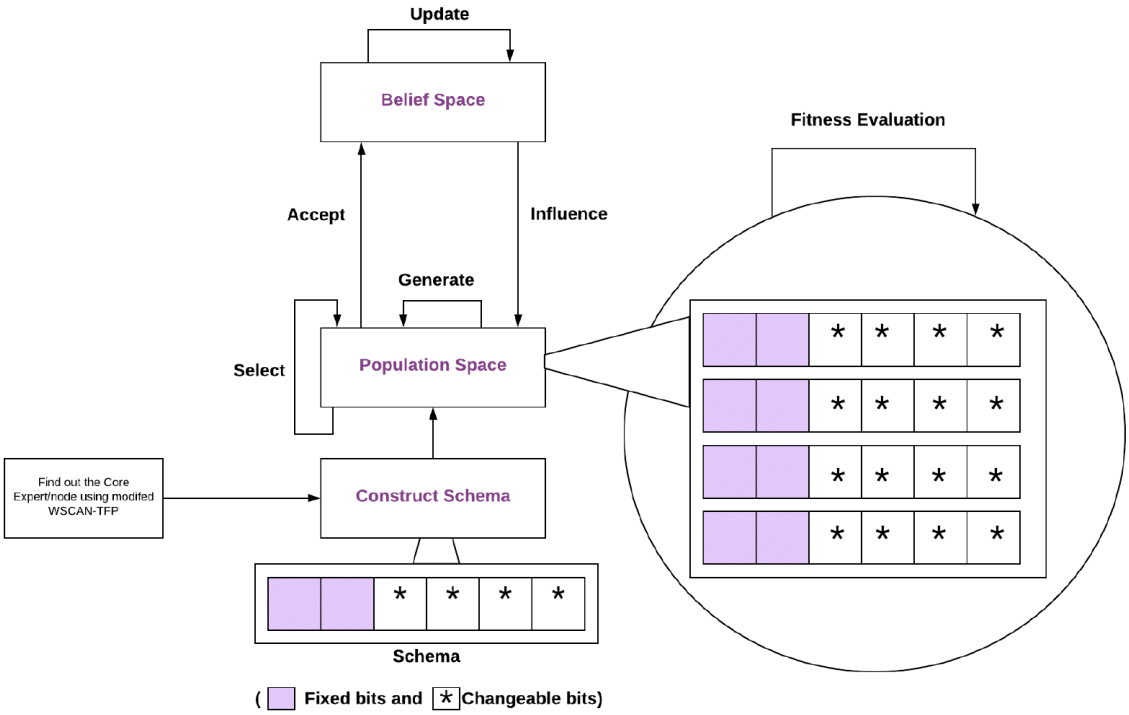


Figure 3.15: Hybrid Cultural Algorithm with Schema.

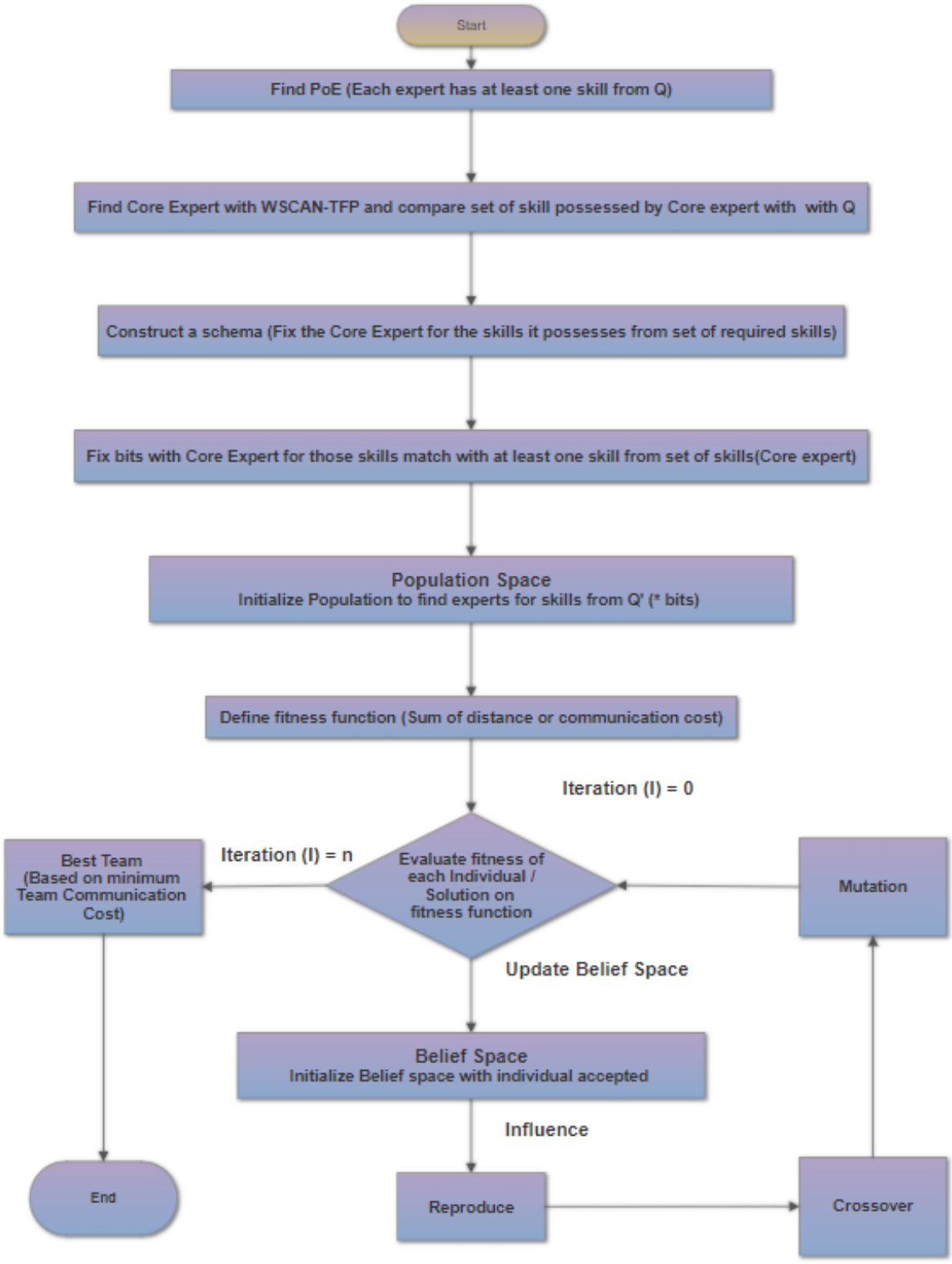


Figure 3.16: Flowchart of Hybrid Cultural Algorithm.

In the third phase of the algorithm, it will initialize the population and we define

a fitness function to check the fitness of each individual / solution. In addition to this belief space is initialized at iteration (I) = 0 and this belief space will be updated with each iteration (I). Iteration is predefined from 0 to N. Process of reproduction, crossover, and mutation is stopped when iteration (I) is equal to N. Termination condition is predefined as in this case, keep on increasing iteration number by one i.e. $I \leftarrow I + 1$ until I reach the maximum iteration number I_{max}.

The difference between Strategy S4 and Strategy S5 is the utilization of belief space in Strategy S5 with the similar process followed in Strategy S4. In the hybrid version of the genetic algorithm and cultural algorithm is the utilization of WSCAN-TFP in the first phase of the algorithm. In the second phase utilization of schema theory with initialization of population, reproduction, crossover and mutation process until termination condition matches. Individuals / Solution are evaluated on the fitness function. Generally, a positive value is assigned to each individual which shows its fitness.

Strategy 5 follows the procedure of the hybrid cultural algorithm (HCA) after generating a social network graph. The experimental results are given in chapter 4 and analysis in chapter 5.

1: **Phase 1- Begin**
2: find PoE expert (WSCAN-TFP)
3: find Core expert (WSCAN-TFP)
4: **Phase1- End**
5:
6: **Phase2- Begin**
7: Search project specific skills R for Core
8: Assign bit string in schema H
9: Fix Core $[Core]_{p(i,j)}, *, *, *$ in H
10: fill * string bits from $Q_k \in Q$; $L \supseteq Q$
11: **if** p_i and p_j same skill as mentioned in Q
12: **else** choose randomly out of above two
13: $[Core]_{p(i,j)}, p_(i, j) \in H$
14: **Phase2- End**
15:
16: **Phase3- Begin**
17: **generate initial population** with H
18: current generation = 1
19: **for** $p(i, j) = 1$ to t **do**
20: evaluate a population
21: for $p_(i, j) = 1$ to t **do**
22: Evaluate $T(p_(i, j))$ on $f(H, t)$
23: **initialize the Belief space**
24: initialize iteration (I) = 0
25: **Selection**
26: **Crossover** for $p_(i, j) = 1$ to t **do**
27: if $(j <= crossoverpoint)$
28: offspring $[i][j]$ = parents $[i][j]$
29: offspring $[i + 1][j]$ = parents $[i + 1][j]$
30: **else**
31: offspring $[i][j]$ = parents $[i + 1][j]$
32: offspring $[i + 1][j]$ = parents $[i][j]$
33: **mutation**
34: for $p_(i, j) = 1$ to t do
35: if random number ¡ mutation rate
36: mutation $(offspring[p_i])$
37: evaluate the offspring
38: **update belief space** with accepted individuals
39: **new population** (influence of belief space)
40: current iteration = I+1
41: $popl$ = offspring
42: current generation + = t
43: evaluate on $f(H, t)$ at generation t
44:
45: **termination condition**
46: current iteration (I) = N
47: **best team** $T([Core]_p(i, j), p_(i, j))$
48: ends HCA
49: **Phase3- End**

Algorithm 4: Hybrid Cultural Algorithm(HCA) with Schema

Chapter 4

Experiments

In this chapter, we are describing the details of the experimental setup. Later in chapter V, we will summarize the results and analyze it.

4.1 Experimental Setup

All the experiments were conducted on a PC with device specifications a 3.40GHz Intel CORE i7-6700 processor, 8 GB of RAM and 64-bit operating system. We implemented all strategies in JDK 9.0.4 (windows-64bit), IntelliJ IDEA 2017.3.4.

To test our strategies on a real network, we are using the DBLP [1] dataset which is one of the expert's network also used in [31], [27], [42] and [41]. The basic concept of DBLP network is that when two authors publish any paper together, they will have the connection between them.

In order to conduct the experiment to test all five strategies, we are considering various attributes and making combinations to test our approach. Attributes for the following experiments are:

[1] *http://dblp.uni-trier.de/xml/*

Combinations	Graph Size	Graph Weight
Combination 1 (C1)	50 K	Equally weighted graph (G1)
Combination 2 (C2)	50 K	Logarithmically weighted graph (G2)
Combination 3 (C3)	50 K	Semantically weighted graph (G3)
Combination 4 (C4)	100 K	Equally weighted graph (G1)
Combination 5 (C5)	100 K	Logarithmically weighted graph (G2)
Combination 6 (C6)	100 K	Semantically weighted graph (G3)

Table 4.1: Different datasets from DBLP network

- Size of the graph: 50K and 100K
- Edge Weight of the graph: equally weighted, logarithmically weighted and semantically weighted
- Strategies: S1, S2, S3, S4, and S5
- Number of required skills: various (3,5, 7, 9, 10, 11, 15, 25, and 30)

All the five strategies explained in chapter 3 were tested in this chapter with various combinations mentioned in table 4.1.

4.2 Methods to generate edge weight

We are using different dataset mentioned in table 4.1 to test all different strategies. For all experiments, initially, the social network is generated with edge weights. Every social network graph represents experts connected to each other with some weight. Every expert possesses a set of skills. Weights show the strength of a relationship between experts.

We used differently weighted graphs to measure quality that is Communication Cost, average fitness and average run-time of all five strategies, we are using to solve TFP. Different methods are used to calculate weights between experts. Authors used methods explained below in their work. We are using the same methods to generate 50K and 100K nodes network for DBLP.

- The author in[42] and [41] used expert's network with equally weighted edge graphs to conduct experiments. It used the 50K nodes of equal edge weight graph with 1.0 of weight on all edges. Where 1.0 shows the connection between two experts

- Author In [42] calculated edge weights of expert's network graph using logarithmic formula mentioned below. It used log of degree of each expert or node (p_i) and p_j. Suppose two experts p_i and p_j is

$$(log_2(1 + deg(p_i) + log_2(1 + deg(p_j))/2)$$

  where, $deg(p_i)$ and $deg(p_j)$ is degree of expert p_i and p_j respectively.

- Author in [42]) used Semantically weighted graphs, it was calculated based on number of co-authorship and publications together by experts in DBLP dataset.

4.3 Non-knowledge based Approach

In subsection **??**,we are implementing WSCAN-TFP algorithm with graph G1, G2 and G3 in order to solve TFP. Then we compare experimental results with different heuristics- Cultural Algorithm (CA), Genetic Algorithm (GA), Hybrid Genetic Algorithm (HGA), and Hybrid Cultural Algorithm (HCA). This experiment used the real

data set of DBLP. We have conducted experiments to get results with 50K and 100K nodes network derived from the DBLP dataset. For the application of the WSCAN-TFP function, We use the sum of distance as a communication function to calculate the edge weight between two experts.

4.4 Strategy 1 (S1) on 50K and 100K nodes network

In fig. 4.1, 4.2, 4.3, 4.4, 4.5 and 4.6, shows the experimental results with WSCAN-TFP on different expert's network with a various number of the set of required skills. However, we found that the result always follows the same pattern.

4.4.1 Experimental results for combination 1 (C1) with S1

In fig. 4.1, we are conducting experiments with S1 using G1 graph for 50K nodes network. We are using the different set of required skills. In this fig. 4.1, we are finding communication cost for the team with experts. These experts satisfy all skills from the set of required skills.

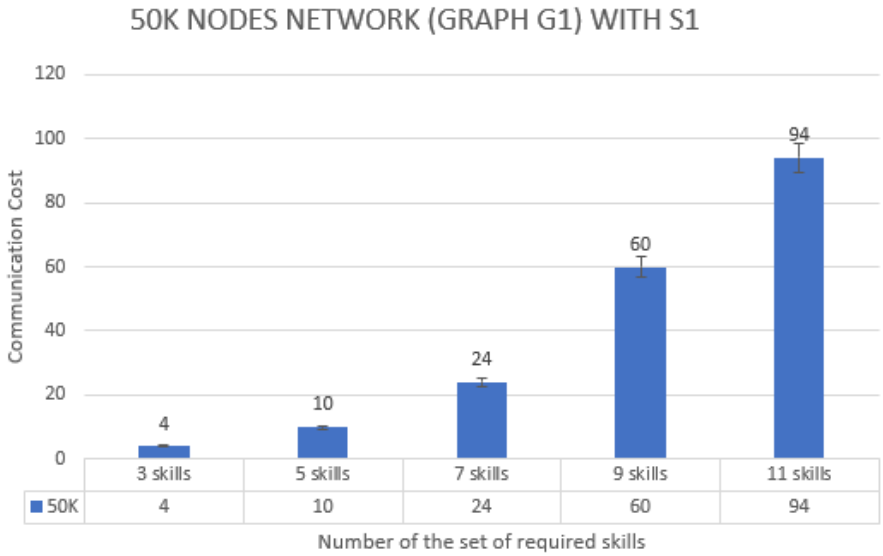


Figure 4.1: S1 on combination 1 (C1): 50k nodes network and G1 type graph

However, we compared WSCAN-TFP with S2, S3, S4, and S5, its performance in terms of quality is disused in detail in chapter 5.

Importantly, the runtime of the WSCAN-TFP later in chapter 5 is very less.

4.4.2 Experimental results for combination 2 (C2) with S1

In fig. 4.2, we are conducting experiments with S1 using G2 graph for 50K nodes network. We are using the different set of required skills. In this fig. 4.2, we are finding communication cost for the team with experts. These experts satisfy all skills from the set of required skills.

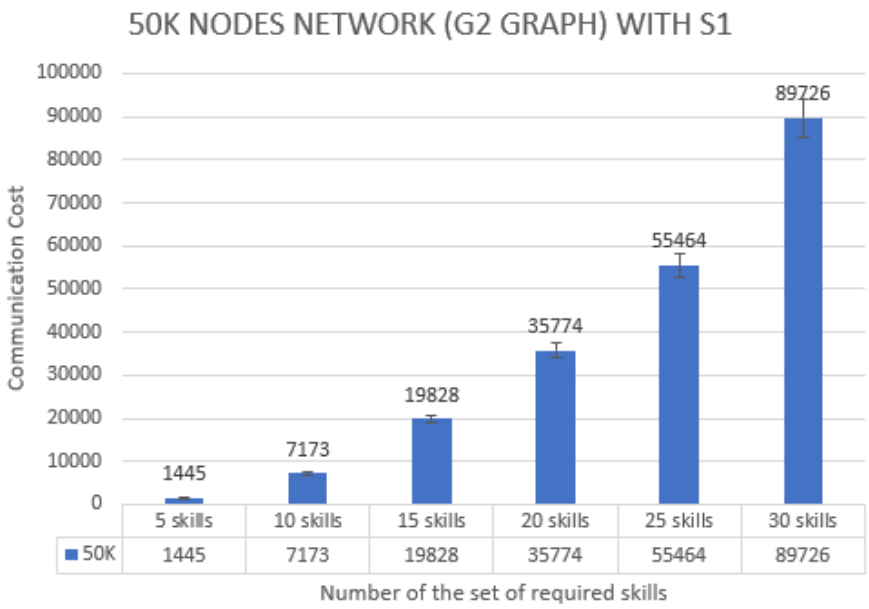


Figure 4.2: S1 on combination 2 (C2): 50k nodes network and G2 type graph

4.4.3 Experimental results for combination 3 (C3) with S1

In fig. 4.3, we are conducting experiments with S1 using G3 graph for 50K nodes network. We are using the different set of required skills. In this fig. 4.3, we are finding communication cost for the team with experts. These experts satisfy all skills from the set of required skills.

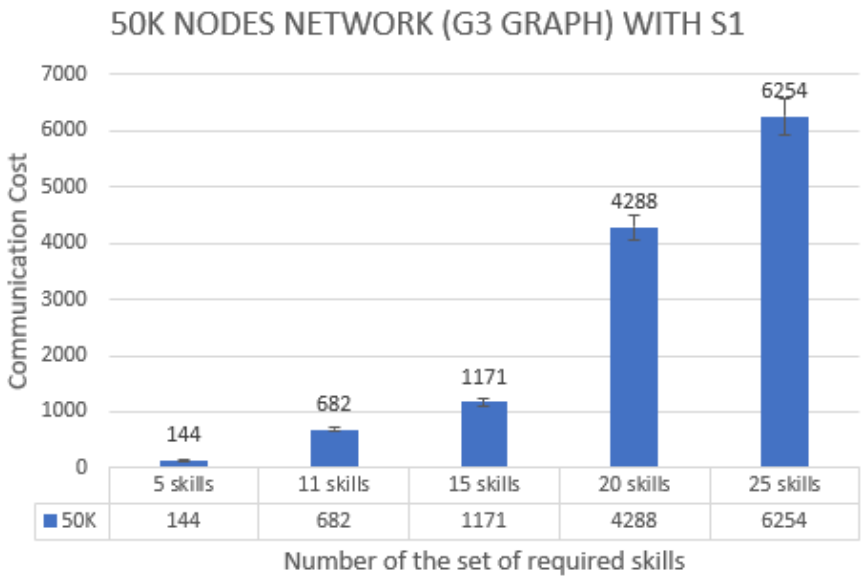


Figure 4.3: S1 on combination 3 (C3): 50k nodes network and G3 type graph

4.4.4 Experimental results for combination 4 (C4) with S1

In fig. 4.4, we are conducting experiments with S1 using G1 graph for 100K nodes network. We are using the different set of required skills. In this fig. 4.4, we are finding communication cost for the team with experts. These experts satisfy all skills from the set of required skills.

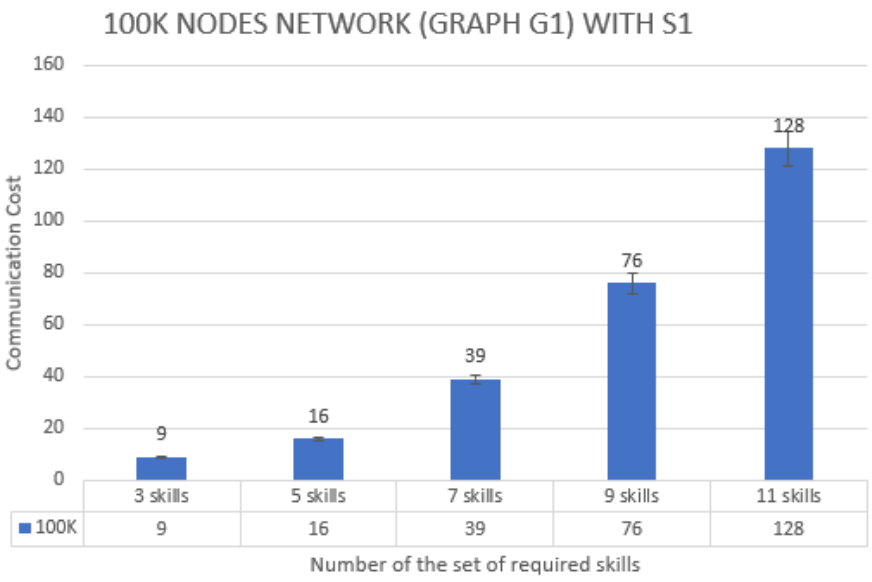


Figure 4.4: S1 on combination 4 (C4): 100k nodes network and G1 type graph

4.4.5 Experimental results for combination 5 (C5) with S1

In fig. 4.5, we are conducting experiments with S1 using G2 graph for 100K nodes network. We are using the different set of required skills. In this fig. 4.5, we are finding communication cost for the team with experts. These experts satisfy all skills from the set of required skills.

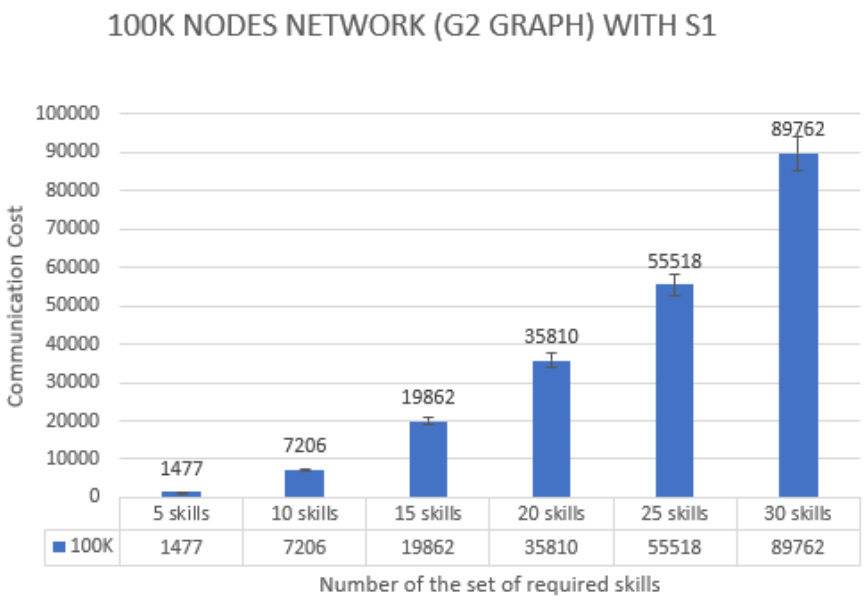


Figure 4.5: S1 on combination 5 (C5): 100k nodes network and G2 type graph

4.4.6 Experimental results for combination 6 (C6) with S1

In fig. 4.6, we are conducting experiments with S1 using G3 graph for 100K nodes network. We are using the different set of required skills. In this fig. 4.6, we are finding communication cost for the team with experts. These experts satisfy all skills from the set of required skills.

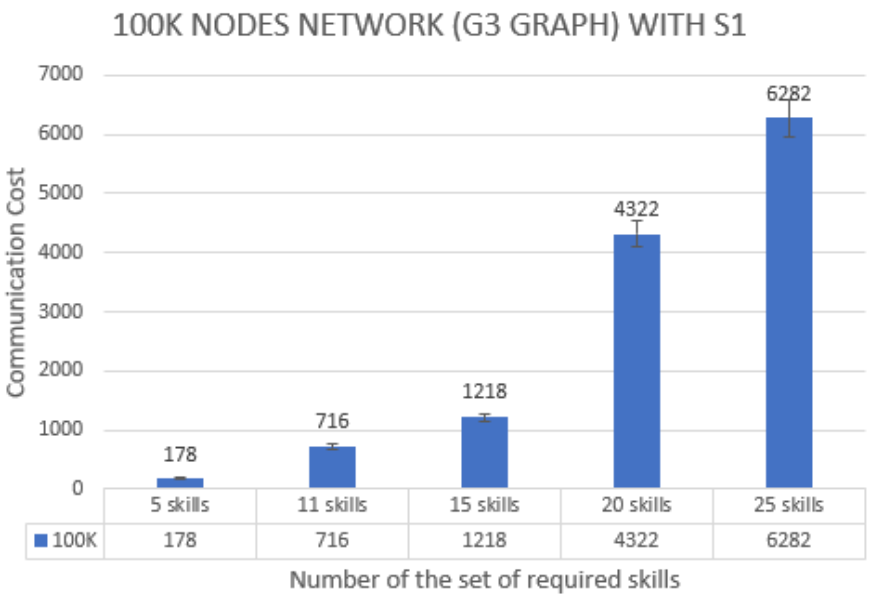


Figure 4.6: S1 on combination 6 (C6): 100k nodes network and G3 type graph

4.5 Knowledge based Approach

In the knowledge-based approach, We are using genetic and cultural algorithms. GA and CA utilize domain knowledge to solve the problem. Apart from existing heuristic mentioned in subsection 4.6 and 4.7, we are proposing two new hybrid heuristic explained in detail in subsection 4.8 and 4.9.

4.6 Strategy 2 (S2) on 50K and 100K nodes network

We are using combinations as explained in table 4.1 to test strategy $S2$.

4.6.1 Experimental results for combination 1 (C1) with S2

In fig. 4.7, we are conducting experiments with S2 using G1 graph for 50K nodes network. We are using the different set of required skills. In this fig. 4.7, we are

finding communication cost for the team with experts. These experts satisfy all skills from the set of required skills.

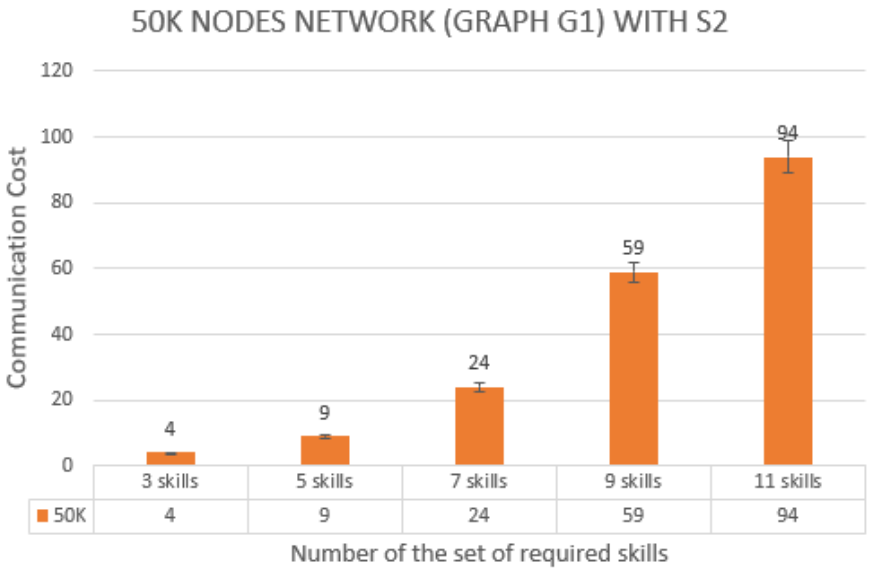


Figure 4.7: S2 on combination 1 (C1): 50k nodes network and G1 type graph

4.6.2 Experimental results for combination 2 (C2) with S2

In fig. 4.8, we are conducting experiments with S1 using G3 graph for 100K nodes network. We are using the different set of required skills. In this fig. 4.8, we are finding communication cost for the team with experts. These experts satisfy all skills from the set of required skills.

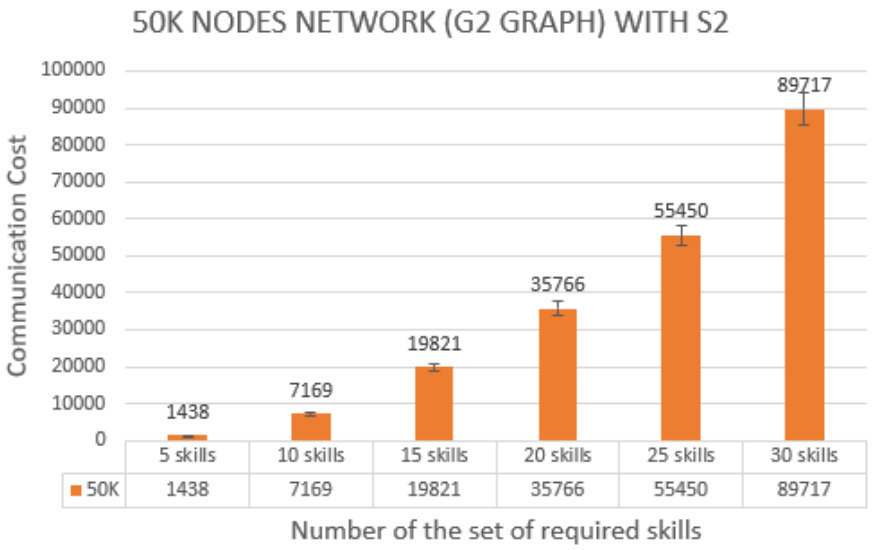


Figure 4.8: S2 on combination 2 (C2): 50k nodes network and G2 type graph

4.6.3 Experimental results for combination 3 (C3) with S2

In fig. 4.9, we are conducting experiments with S2 using G3 graph for 50K nodes network. We are using the different set of required skills. In this fig. 4.9, we are finding communication cost for the team with experts. These experts satisfy all skills from the set of required skills.

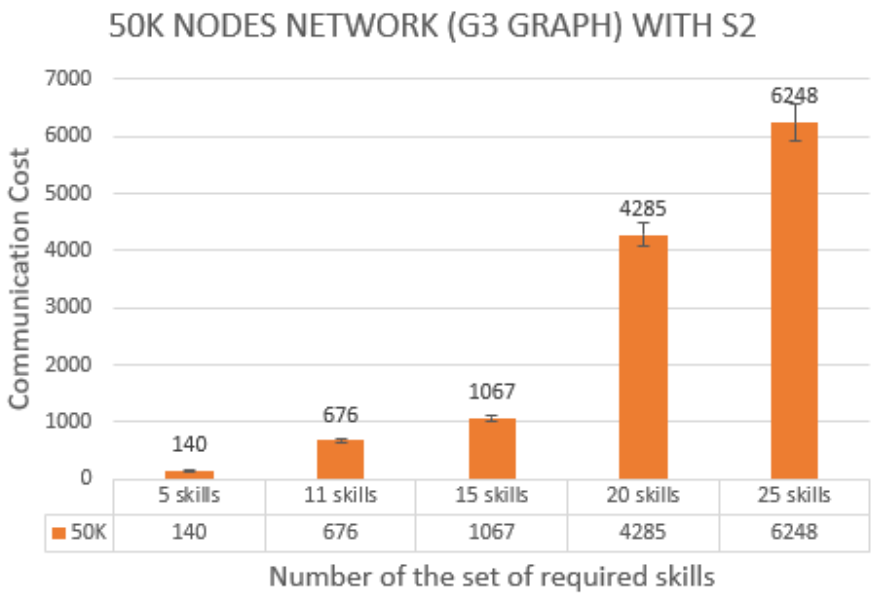


Figure 4.9: S2 on combination 3 (C3): 50k nodes network and G3 type graph

4.6.4 Experimental results for combination 4 (C4) with S2

In fig. 4.10, we are conducting experiments with S2 using G1 graph for 100K nodes network. We are using the different set of required skills. In this fig. 4.10, we are finding communication cost for the team with experts. These experts satisfy all skills from the set of required skills.

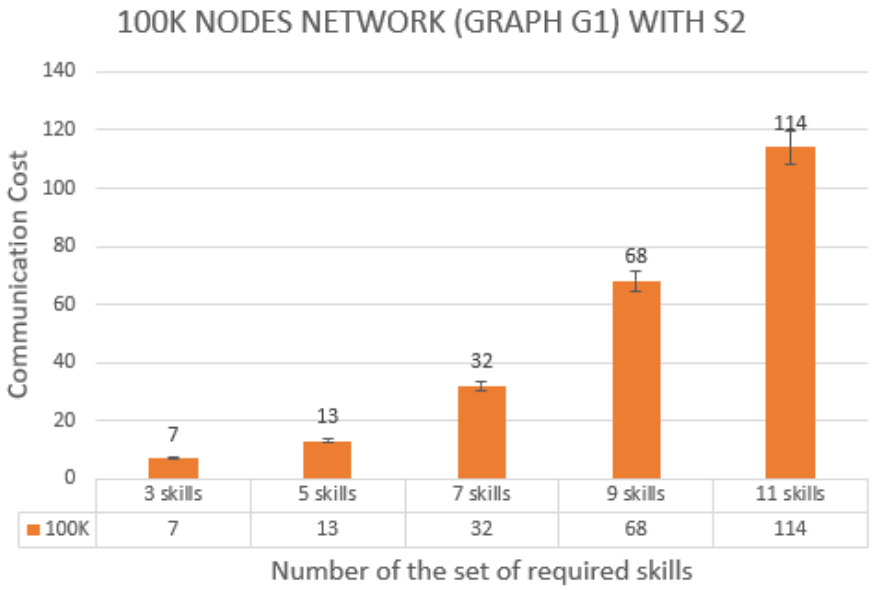


Figure 4.10: S2 on combination 4 (C4): 100k nodes network and G1 type graph

4.6.5 Experimental results for combination 5 (C5) with S2

In fig. 4.11, we are conducting experiments with S2 using G2 graph for 100K nodes network. We are using the different set of required skills. In this fig. 4.11, we are finding communication cost for the team with experts. These experts satisfy all skills from the set of required skills.

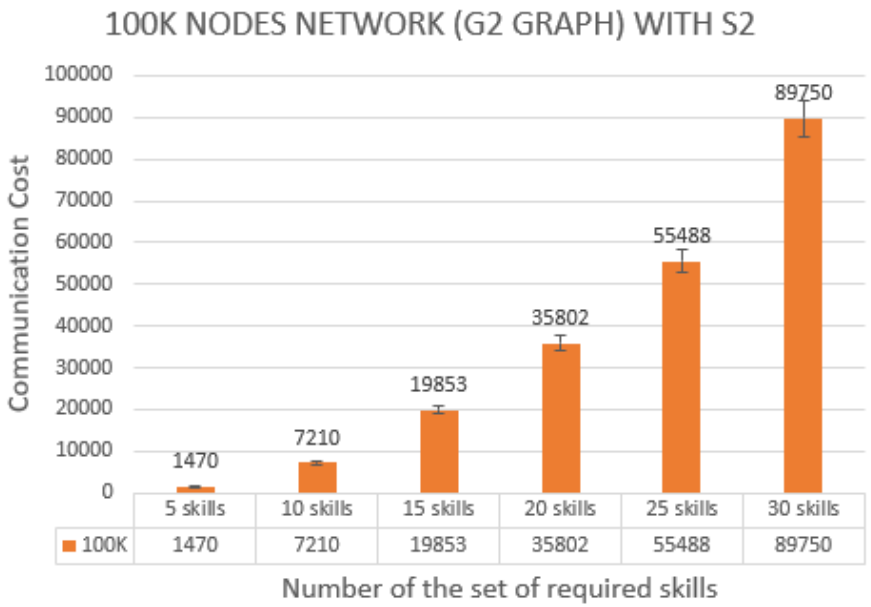


Figure 4.11: S2 on combination 5 (C5): 100k nodes network and G2 type graph

4.6.6 Experimental results for combination 6 (C6) with S2

In fig. 4.12, we are conducting experiments with S2 using G3 graph for 100K nodes network. We are using the different set of required skills. In this fig. 4.12, we are finding communication cost for the team with experts. These experts satisfy all skills from the set of required skills.

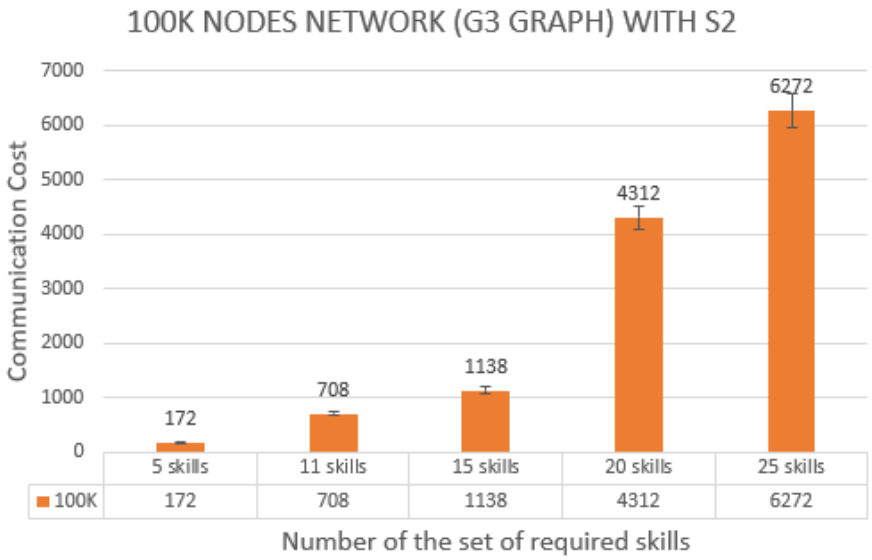


Figure 4.12: S2 on combination 6 (C6): 100k nodes network and G3 type graph

4.7 Strategy 3 (S3) on 50K and 100K nodes network

4.7.1 Experimental results for combination 1 (C1) with S3

In fig. 4.13, we are conducting experiments with S3 using G1 graph for 50K nodes network. We are using the different set of required skills. In this fig. 4.13, we are finding communication cost for the team with experts. These experts satisfy all skills from the set of required skills.

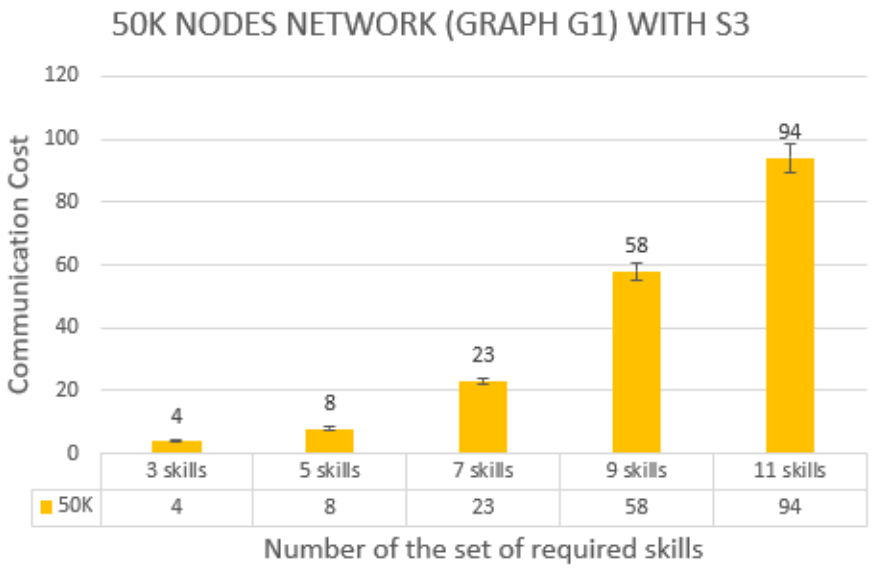


Figure 4.13: S3 on combination 1 (C1): 50k nodes network and G1 type graph

4.7.2 Experimental results for combination 2 (C2) with S3

In fig. 4.14, we are conducting experiments with S3 using G2 graph for 50K nodes network. We are using the different set of required skills. In this fig. 4.14, we are finding communication cost for the team with experts. These experts satisfy all skills from the set of required skills.

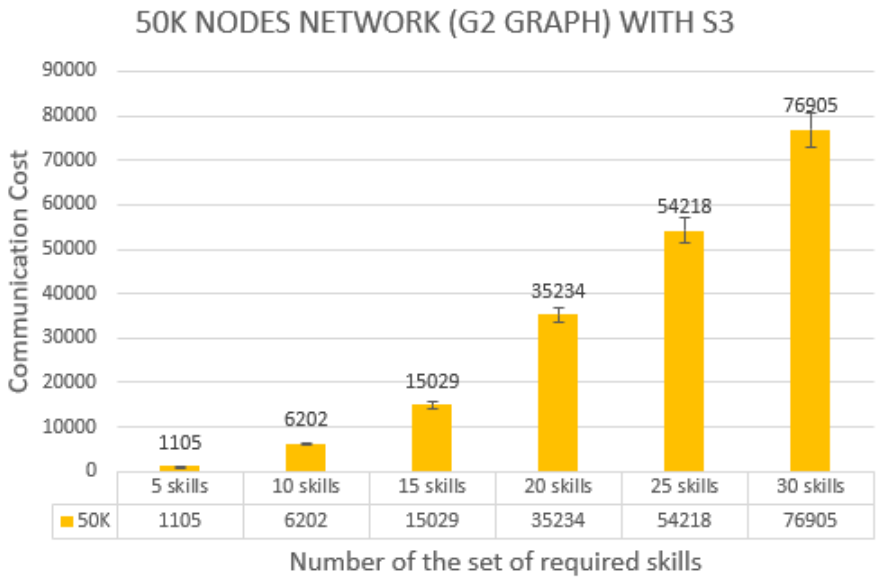


Figure 4.14: S3 on combination 2 (C2): 50k nodes network and G2 type graph

4.7.3 Experimental results for combination 3 (C3) with S3

In fig. 4.15, we are conducting experiments with S3 using G3 graph for 50K nodes network. We are using the different set of required skills. In this fig. 4.15, we are finding communication cost for the team with experts. These experts satisfy all skills from the set of required skills.

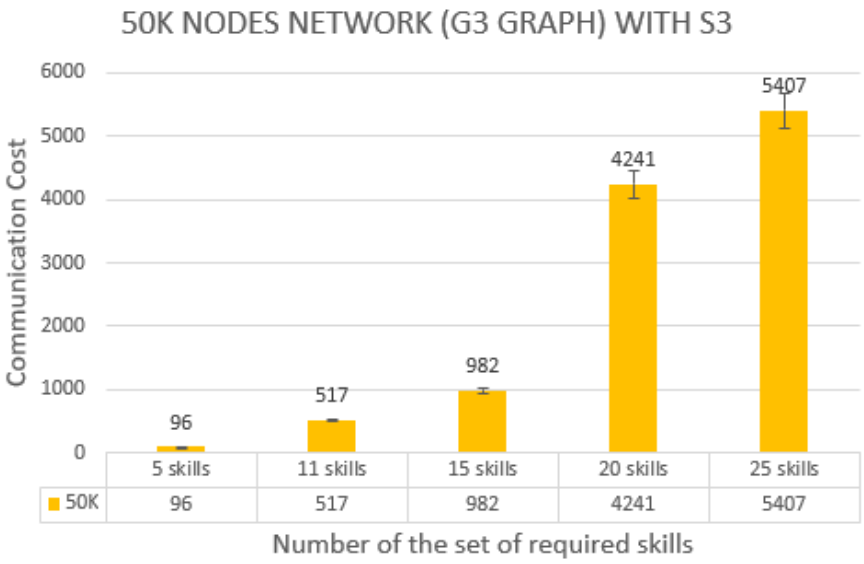


Figure 4.15: S3 on combination 3 (C3): 50k nodes network and G3 type graph

4.7.4 Experimental results for combination 4 (C4) with S3

In fig. 4.16, we are conducting experiments with S3 using G1 graph for 100K nodes network. We are using the different set of required skills. In this fig. 4.16, we are finding communication cost for the team with experts. These experts satisfy all skills from the set of required skills.

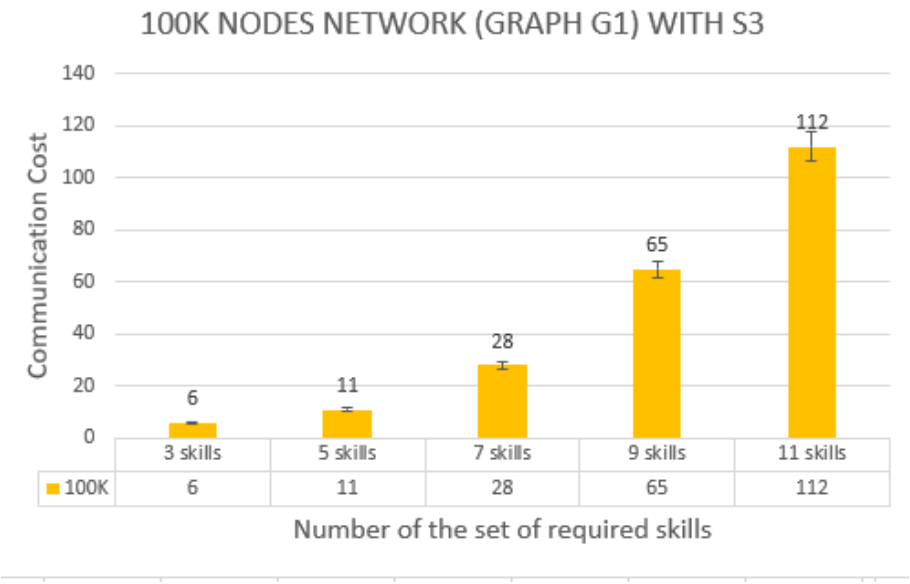


Figure 4.16: S3 on combination 4 (C4): 100k nodes network and G1 type graph

4.7.5 Experimental results for combination 5 (C5) with S3

In fig. 4.17, we are conducting experiments with S3 using G2 graph for 100K nodes network. We are using the different set of required skills. In this fig. 4.17, we are finding communication cost for the team with experts. These experts satisfy all skills from the set of required skills.

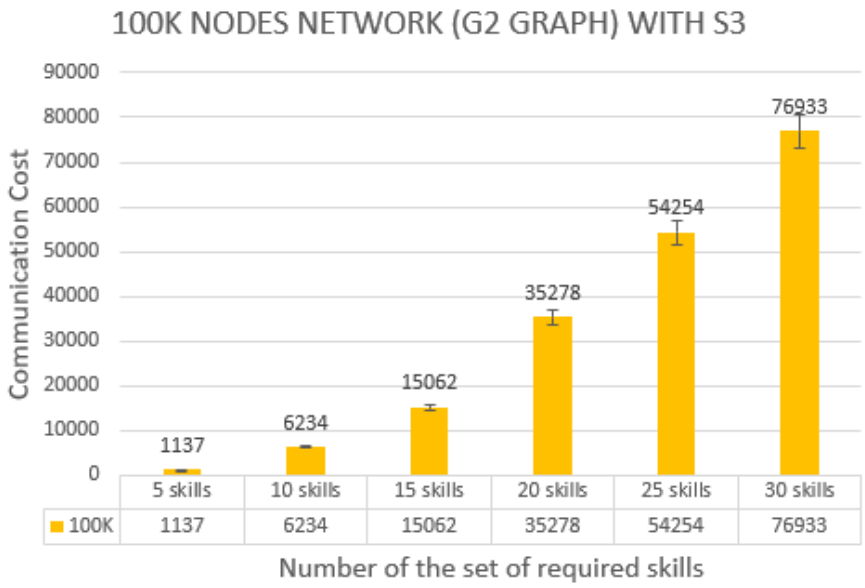


Figure 4.17: S3 on combination 5 (C5): 100k nodes network and G2 type graph

4.7.6 Experimental results for combination 6 (C6) with S3

In fig. 4.18, we are conducting experiments with S3 using G3 graph for 100K nodes network. We are using the different set of required skills. In this fig. 4.18, we are finding communication cost for the team with experts. These experts satisfy all skills from the set of required skills.

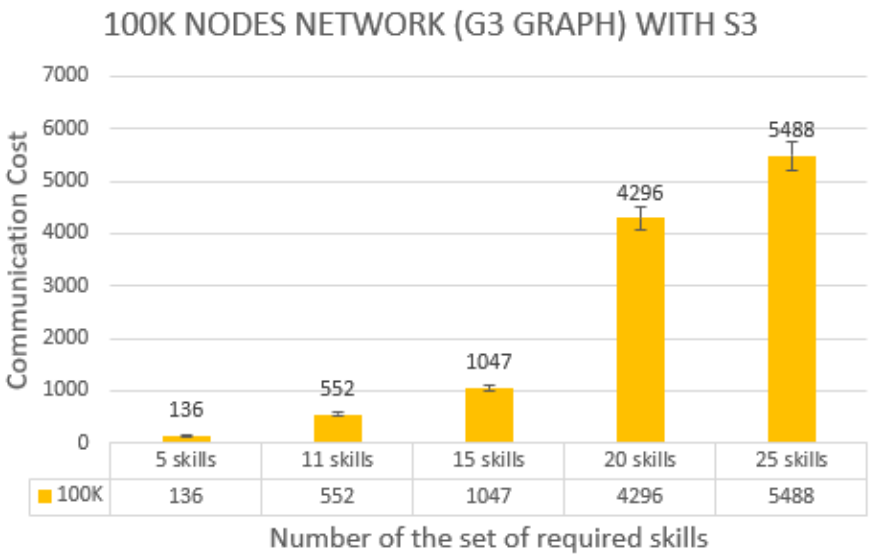


Figure 4.18: S3 on combination 6 (C6): 100k nodes network and G3 type graph

4.8 Strategy 4 (S4) on 50K and 100K nodes network

4.8.1 Experimental results for combination 1 (C1) with S4

In fig. 4.19, we are conducting experiments with S4 using G1 graph for 50K nodes network. We are using the different set of required skills. In this fig. 4.19, we are finding communication cost for the team with experts. These experts satisfy all skills from the set of required skills.

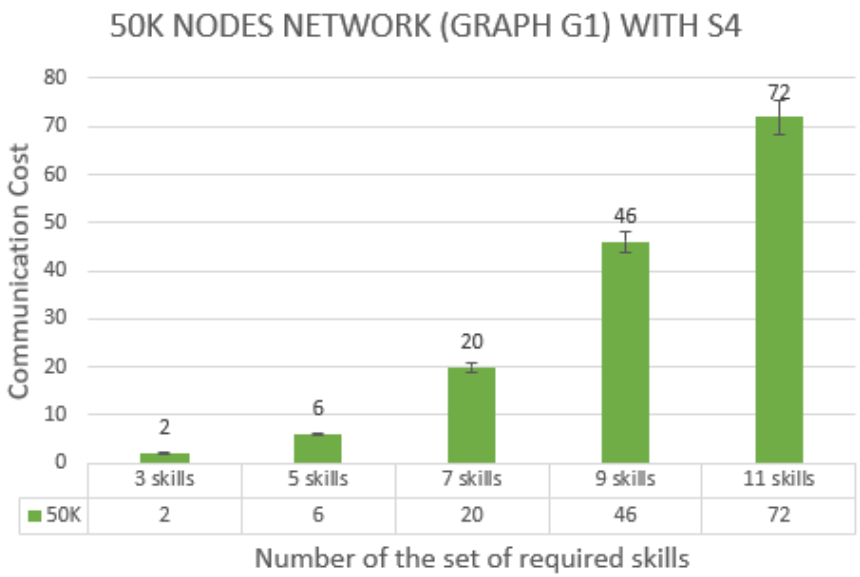


Figure 4.19: S4 on combination 1 (C1): 50k nodes network and G1 type graph

4.8.2 Experimental results for combination 2 (C2) with S4

In fig. 4.20, we are conducting experiments with S4 using G2 graph for 50K nodes network. We are using the different set of required skills. In this fig. 4.20, we are finding communication cost for the team with experts. These experts satisfy all skills from the set of required skills.

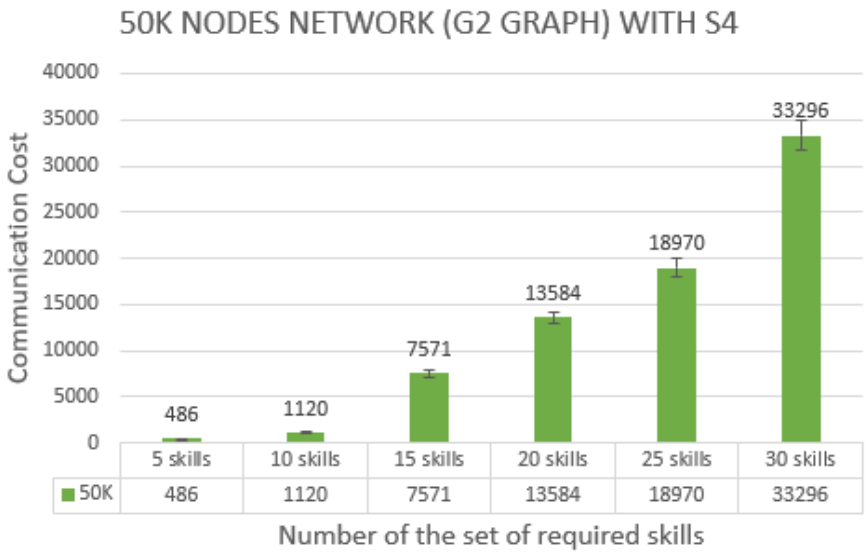


Figure 4.20: S4 on combination 2 (C2): 50k nodes network and G2 type graph

4.8.3 Experimental results for combination 3 (C3) with S4

In fig. 4.21, we are conducting experiments with S4 using G3 graph for 50K nodes network. We are using the different set of required skills. In this fig. 4.21, we are finding communication cost for the team with experts. These experts satisfy all skills from the set of required skills.

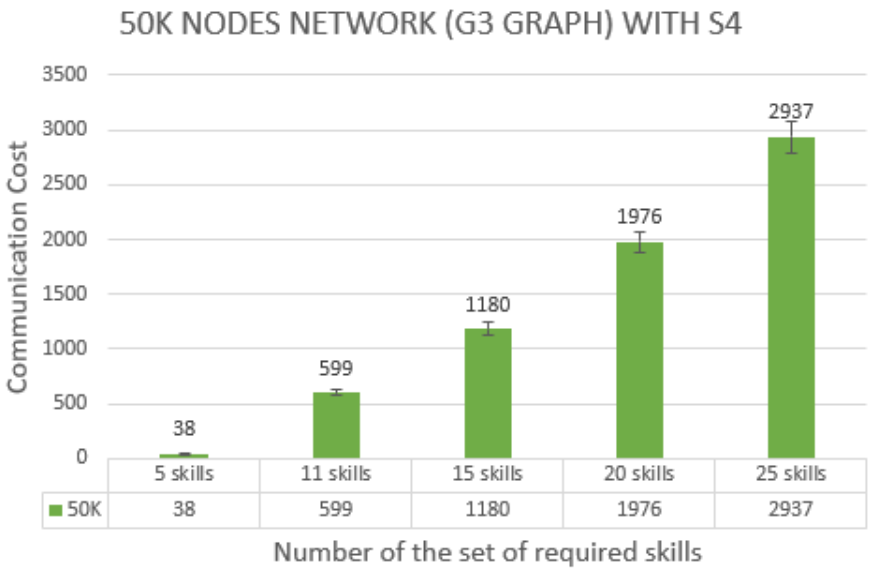


Figure 4.21: S4 on combination 3 (C3): 50k nodes network and G3 type graph

4.8.4 Experimental results for combination 4 (C4) with S4

In fig. 4.22, we are conducting experiments with S4 using G1 graph for 100K nodes network. We are using the different set of required skills. In this fig. 4.22, we are finding communication cost for the team with experts. These experts satisfy all skills from the set of required skills.

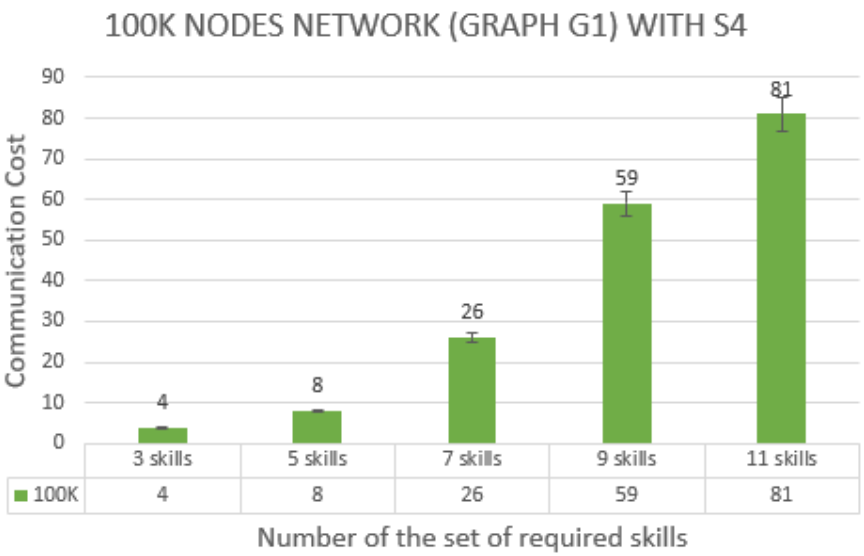


Figure 4.22: S4 on combination 4 (C4): 100k nodes network and G1 type graph

4.8.5 Experimental results for combination 5 (C5) with S4

In fig. 4.23, we are conducting experiments with S4 using G2 graph for 100K nodes network. We are using the different set of required skills. In this fig. 4.23, we are finding communication cost for the team with experts. These experts satisfy all skills from the set of required skills.

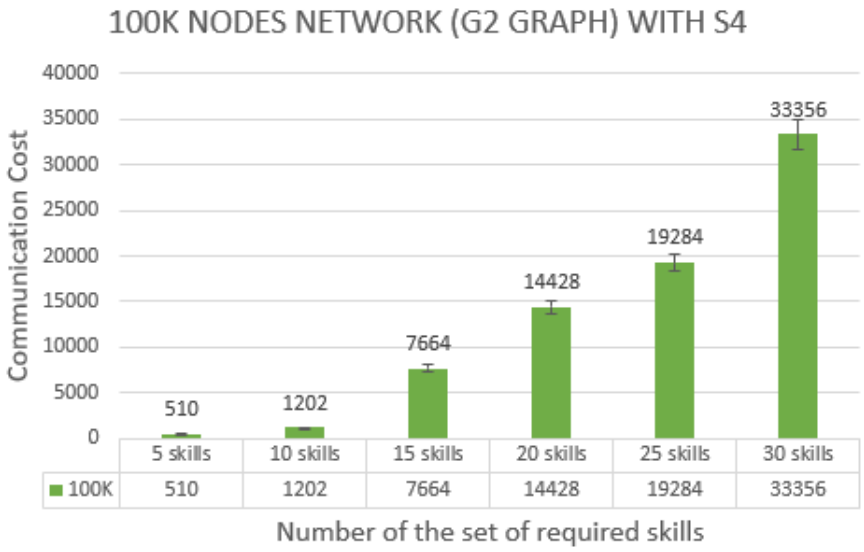


Figure 4.23: S4 on combination 5 (C5): 100k nodes network and G2 type graph

4.8.6 Experimental results for combination 6 (C6) with S4

In fig. 4.24, we are conducting experiments with S4 using G3 graph for 100K nodes network. We are using the different set of required skills. In this fig. 4.24, we are finding communication cost for the team with experts. These experts satisfy all skills from the set of required skills.

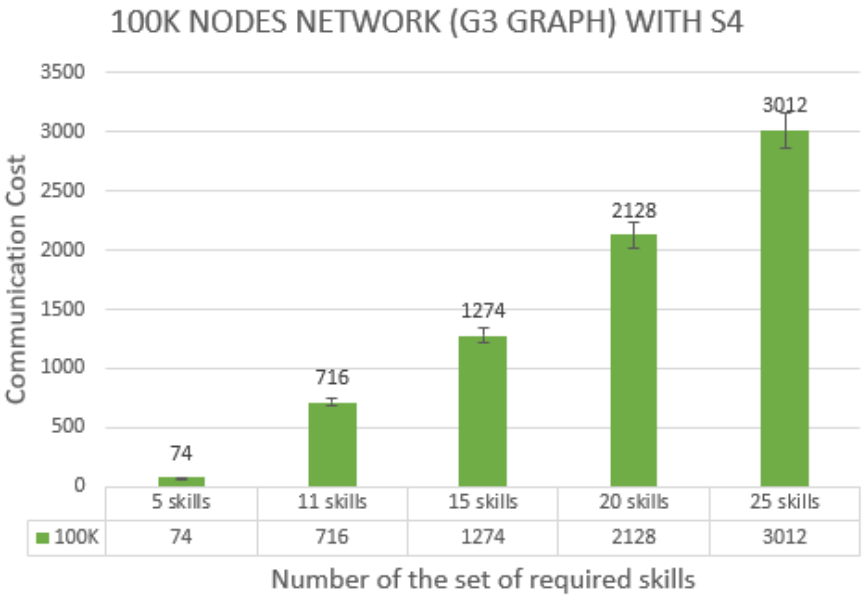


Figure 4.24: S4 on combination 6 (C6): 100k nodes network and G3 type graph

We tested with different attributes , but the trend of increase in communication cost with an increase in the number of required skills for the team remains same.

4.9 Strategy 5 (S5) on 50K and 100K nodes network

4.9.1 Experimental results for combination 1 (C1) with S5

In fig. 4.25, we are conducting experiments with S5 using G1 graph for 50K nodes network. We are using the different set of required skills. In this fig. 4.25, we are

finding communication cost for the team with experts. These experts satisfy all skills from the set of required skills.

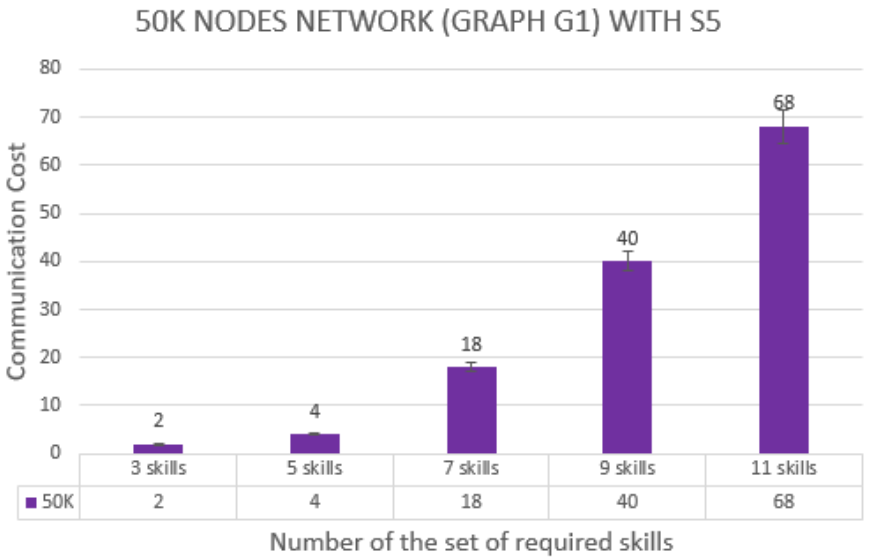


Figure 4.25: S5 on combination 1 (C1): 50k nodes network and G1 type graph

4.9.2 Experimental results for combination 2 (C2) with S5

In fig. **??**, we are conducting experiments with S5 using G2 graph for 50K nodes network. We are using the different set of required skills. In this fig. **??**, we are finding communication cost for the team with experts. These experts satisfy all skills from the set of required skills.

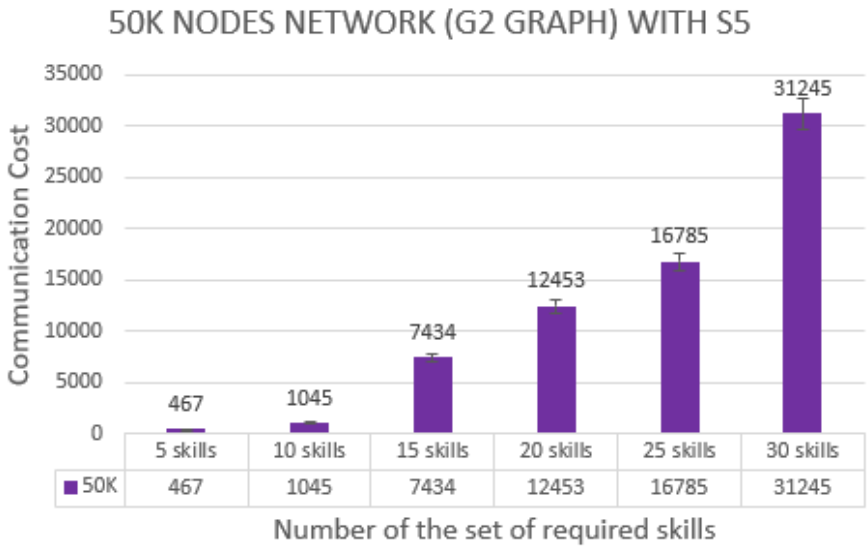


Figure 4.26: S5 on combination 2 (C2): 50k nodes network and G2 type graph

4.9.3 Experimental results for combination 3 (C3) with S5

In fig. 4.27, we are conducting experiments with S5 using G3 graph for 50K nodes network. We are using the different set of required skills. In this fig. 4.27, we are finding communication cost for the team with experts. These experts satisfy all skills from the set of required skills.

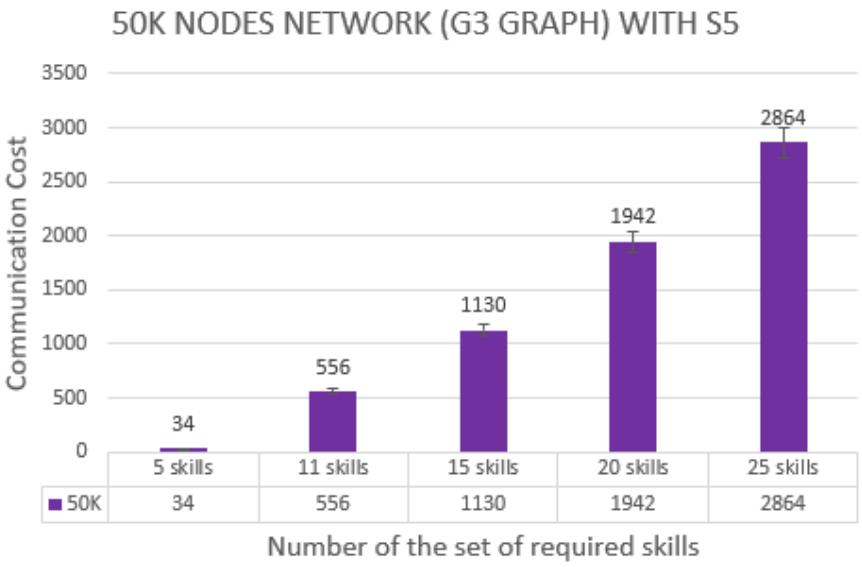


Figure 4.27: S5 on combination 3 (C3): 50k nodes network and G3 type graph

4.9.4 Experimental results for combination 4 (C4) with S5

In fig. 4.28, we are conducting experiments with S5 using G1 graph for 100K nodes network. We are using the different set of required skills. In this fig. 4.28, we are finding communication cost for the team with experts. These experts satisfy all skills from the set of required skills.

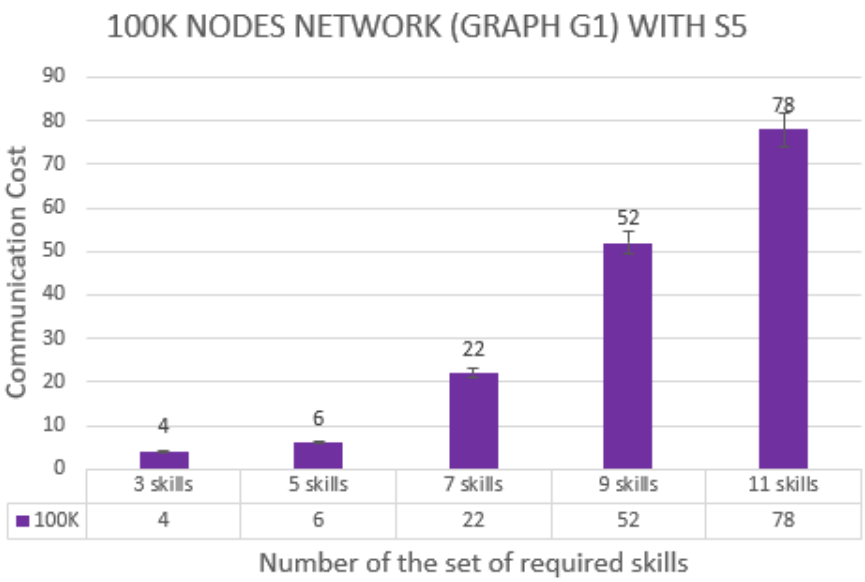


Figure 4.28: S5 on combination 5 (C4): 100k nodes network and G2 type graph

4.9.5 Experimental results for combination 5 (C5) with S5

In fig. 4.29, we are conducting experiments with S5 using G2 graph for 100K nodes network. We are using the different set of required skills. In this fig. 4.29, we are finding communication cost for the team with experts. These experts satisfy all skills from the set of required skills.

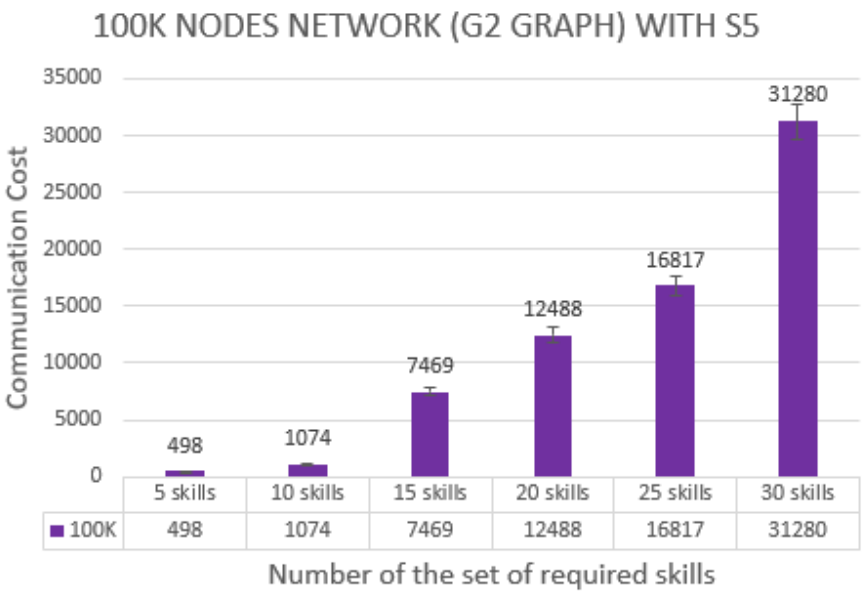


Figure 4.29: S5 on combination 5 (C5): 100k nodes network and G2 type graph

4.9.6 Experimental results for combination 6 (C6) with S5

In fig. 4.30, we are conducting experiments with S5 using G3 graph for 100K nodes network. We are using the different set of required skills. In this fig. 4.30, we are finding communication cost for the team with experts. These experts satisfy all skills from the set of required skills.

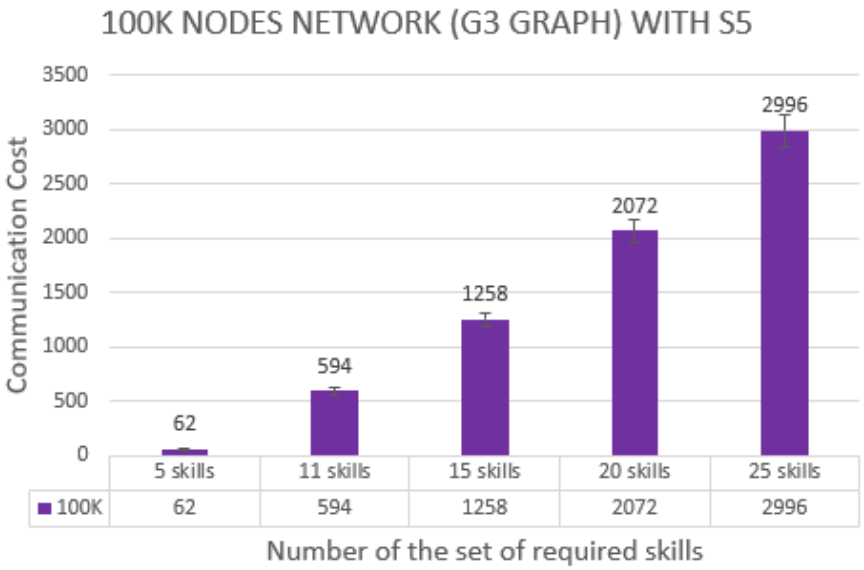


Figure 4.30: S5 on combination 6 (C6): 100k nodes network and G3 type graph

Chapter 5

Discussions, Comparisons and Analysis

In chapter V, we are discussing the results we found after conducting various experiments mentioned in chapter VI.

5.1 Comparison and Analysis

As mentioned in chapter 1, we are focused on the quality of the proposed hybrid strategies. To measure the performance, we are defining three quality measurements. We define these quality measures as follows.

- Communication Cost(CC) with the sum of distance function
- Communication Cost(CC) with diameter of the team
- Average Fitness Score (AFS)
- Average processing time

- Percentage difference

To conduct analysis, we are elaborating our observations based on Communication Cost(CC), Average Fitness Score and average processing time (time-taken) by the heuristic. In order to prove our experiments empirically, we are comparing communication cost, AFS and average processing time (time-taken) in tables and figures.

5.2 Performance measurement with Communication Cost

5.2.1 Communication cost with sum of distance function

Sum of distance can be defined as summation of all the edge weights between team members.

5.2.2 Comparison of C1 with S1, S2, S3, S4 and S5

In fig. 5.1, we are comparing experimental results for S1, S2, S3, S4 and S5 using G1 graph for 50K nodes network. We are using the different set of required skills. In this fig. 5.1, we are finding communication cost for the team with experts. These experts satisfy all skills from the set of required skills.

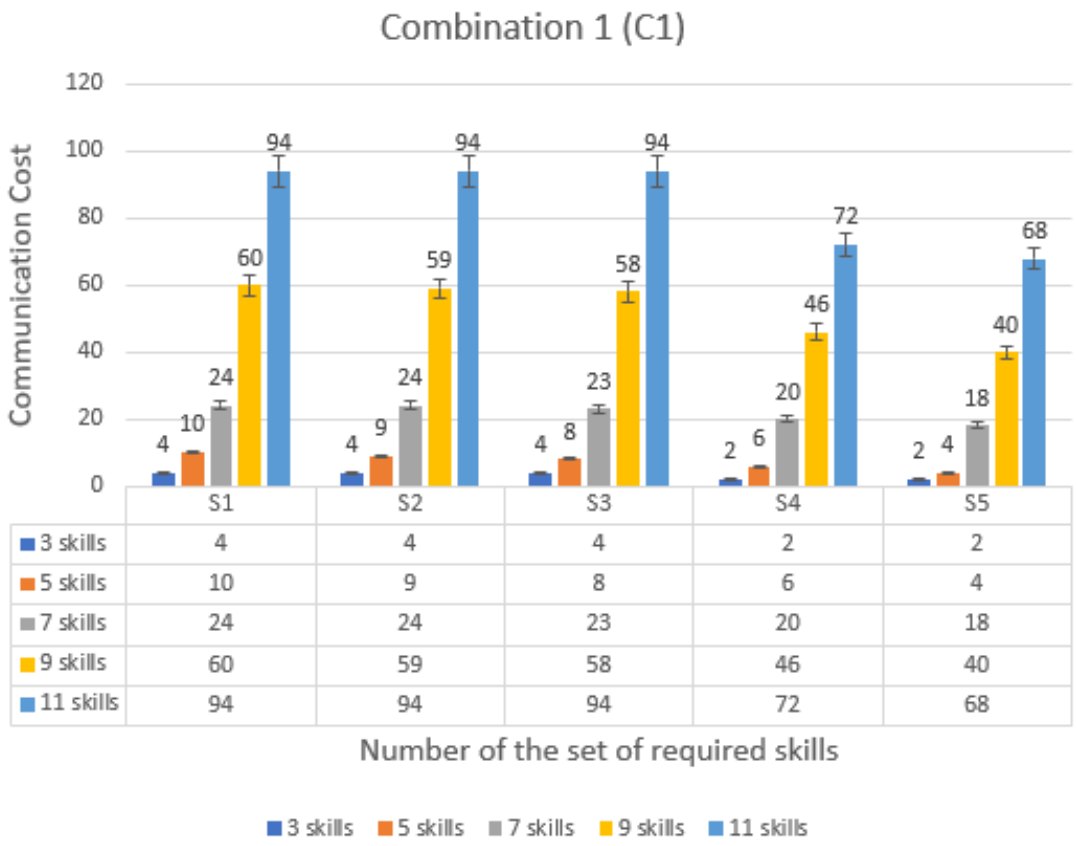


Figure 5.1: communication cost comparison for C1 with S1, S2, S3, S4 and S5

5.2.3 Comparison of C2 with S1, S2, S3, S4 and S5

In fig. 5.2, we are comparing experimental results for S1, S2, S3, S4 and S5 using G2 graph for 50K nodes network. We are using the different set of required skills. In this fig. 5.2, we are finding communication cost for the team with experts. These experts satisfy all skills from the set of required skills.

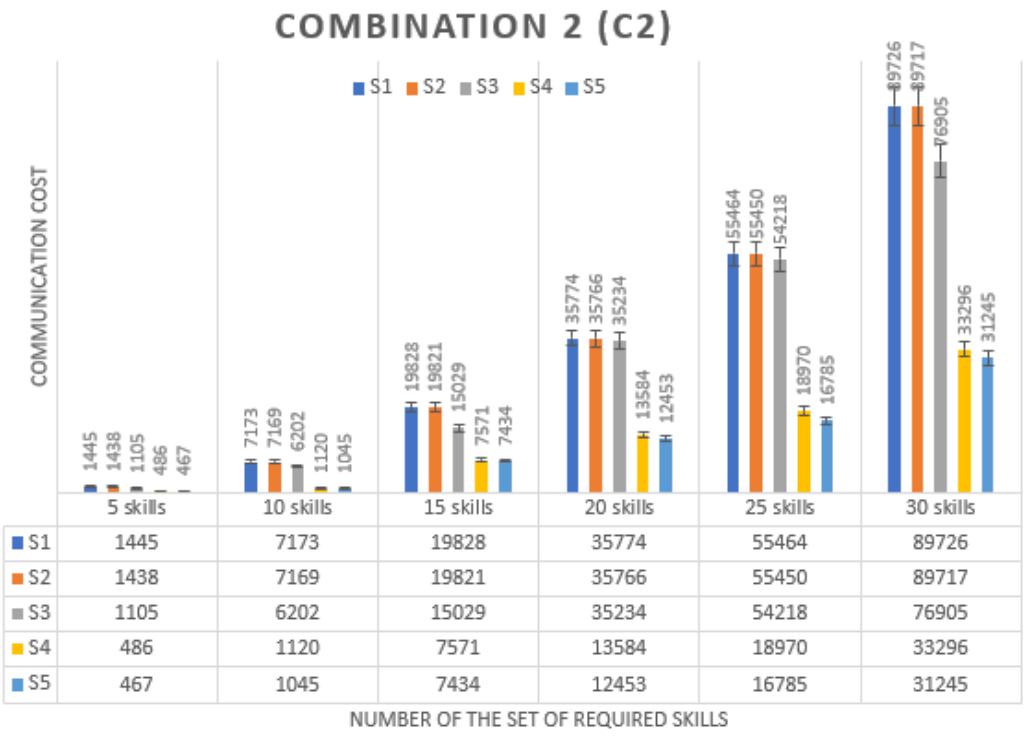


Figure 5.2: communication cost comparison for C2 with S1, S2, S3, S4 and S5

5.2.4 Comparison of C3 with S1, S2, S3, S4 and S5

In fig. 5.3, we are comparing experimental results for S1, S2, S3, S4 and S5 using G3 graph for 50K nodes network. We are using the different set of required skills. In this fig. 5.3, we are finding communication cost for the team with experts. These experts satisfy all skills from the set of required skills.

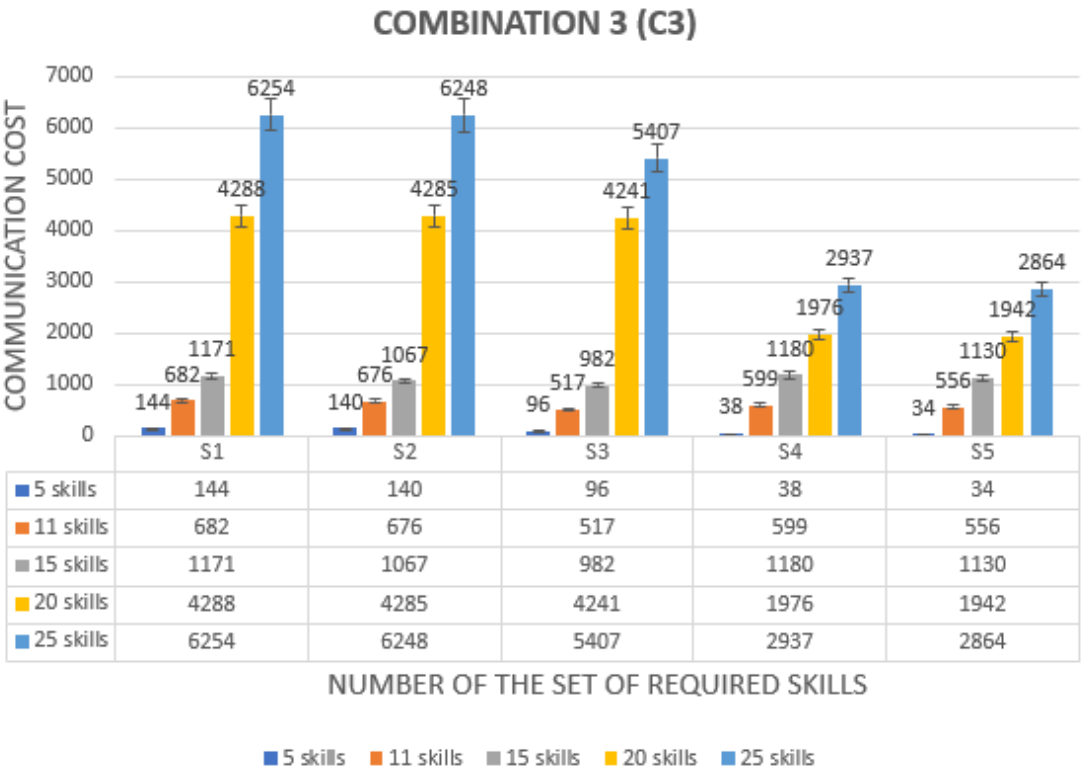


Figure 5.3: communication cost comparison for C3 with S1, S2, S3, S4 and S5

5.2.5 Comparison of C4 with S1, S2, S3, S4 and S5

In fig. 5.4, we are comparing experimental results for S1, S2, S3, S4 and S5 using G1 graph for 100K nodes network. We are using the different set of required skills. In this fig. 5.4, we are finding communication cost for the team with experts. These experts satisfy all skills from the set of required skills.

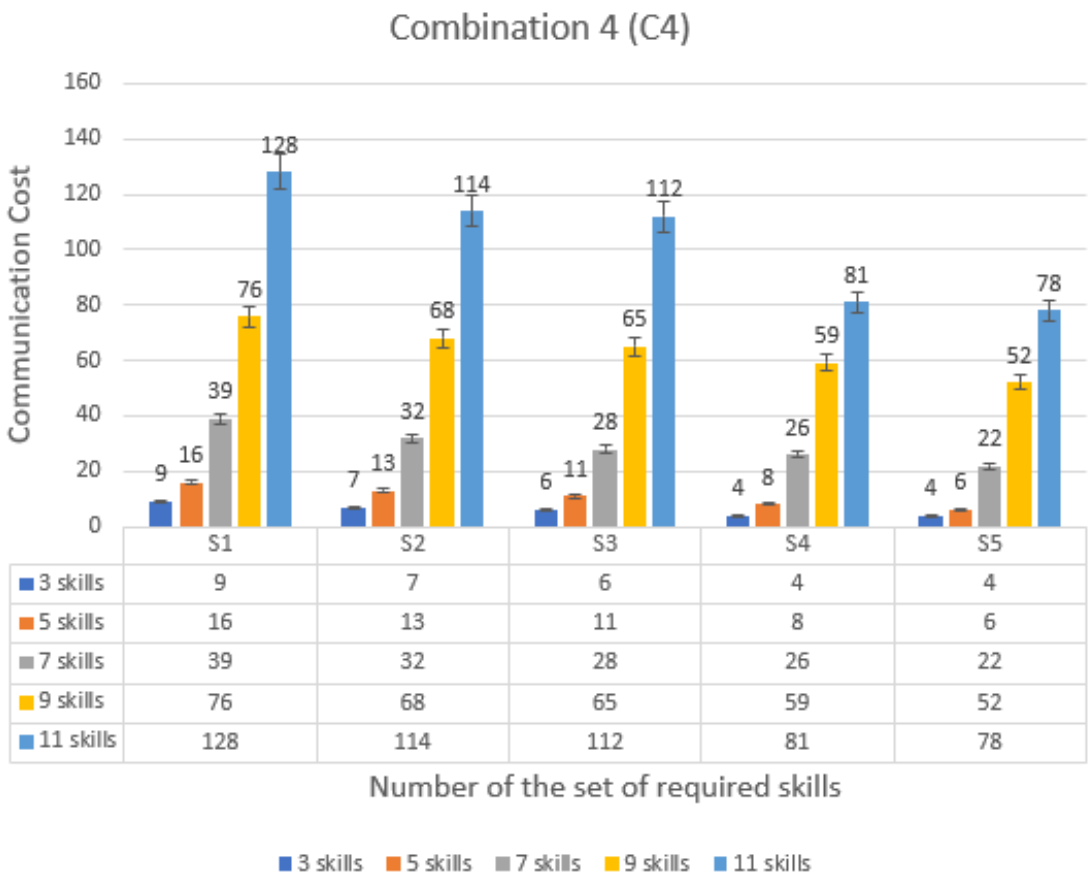
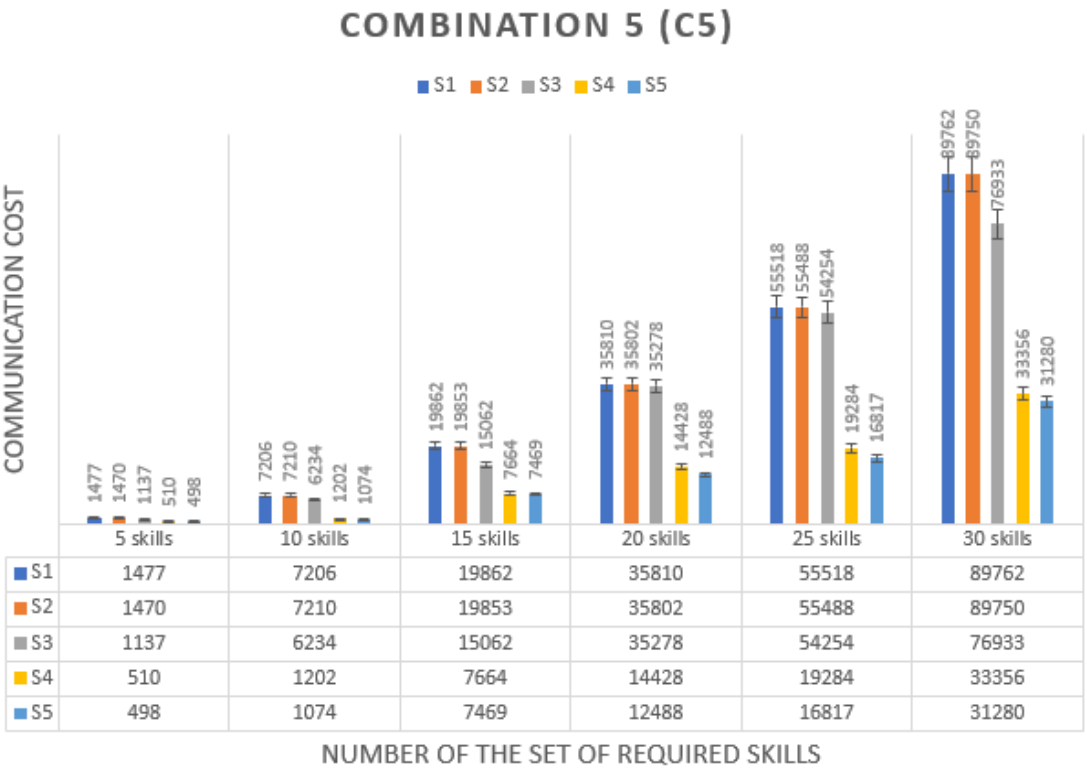


Figure 5.4: communication cost comparison for C4 with S1, S2, S3, S4 and S5

5.2.6 Comparison of C5 with S1, S2, S3, S4 and S5

In fig. 5.5, we are comparing experimental results for S1, S2, S3, S4 and S5 using G2 graph for 100K nodes network. We are using the different set of required skills. In this fig. 5.5, we are finding communication cost for the team with experts. These experts satisfy all skills from the set of required skills.



	5 skills	10 skills	15 skills	20 skills	25 skills	30 skills
S1	1477	7206	19862	35810	55518	89762
S2	1470	7210	19853	35802	55488	89750
S3	1137	6234	15062	35278	54254	76933
S4	510	1202	7664	14428	19284	33356
S5	498	1074	7469	12488	16817	31280

Figure 5.5: communication cost comparison for C5 with S1, S2, S3, S4 and S5

5.2.7 Comparison of C6 with S1, S2, S3, S4 and S5

In fig. 5.6, we are comparing experimental results for S1, S2, S3, S4 and S5 using G3 graph for 100K nodes network. We are using the different set of required skills. In this fig. 5.6, we are finding communication cost for the team with experts. These experts satisfy all skills from the set of required skills.

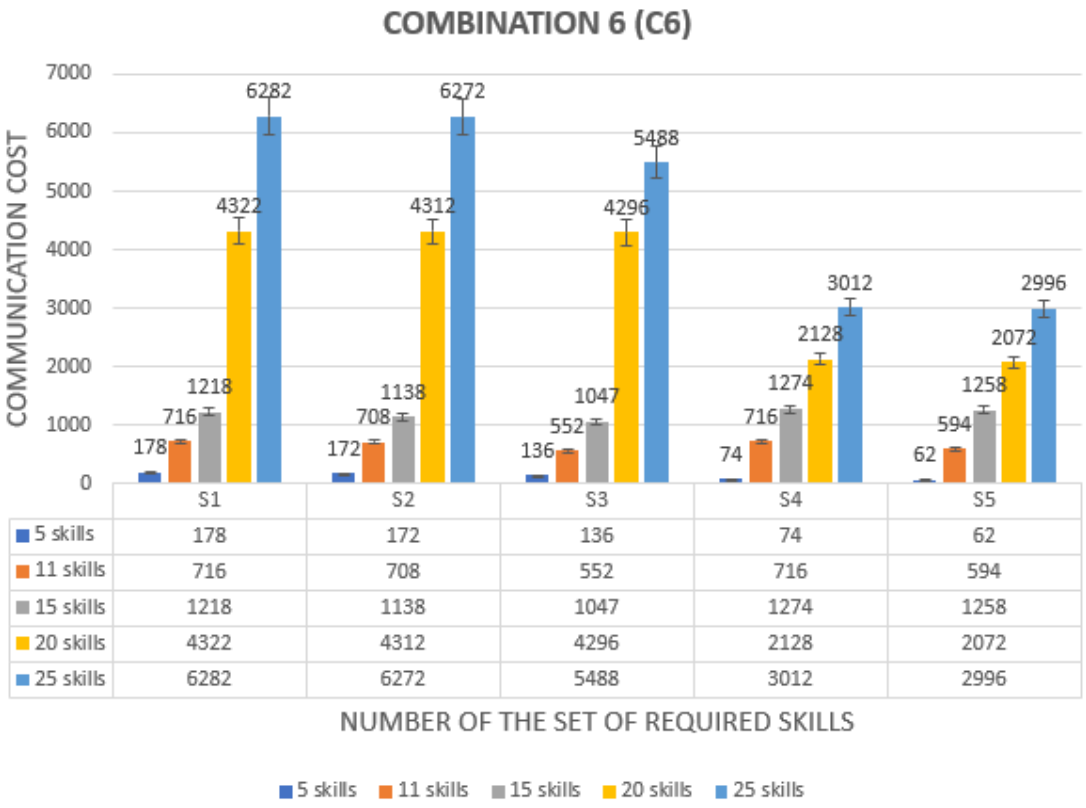


Figure 5.6: communication cost comparison for C6 with S1, S2, S3, S4 and S5

5.2.8 Communication cost with diameter function

The diameter function only measures the communication cost between the two experts that are furthest away from each other [26].

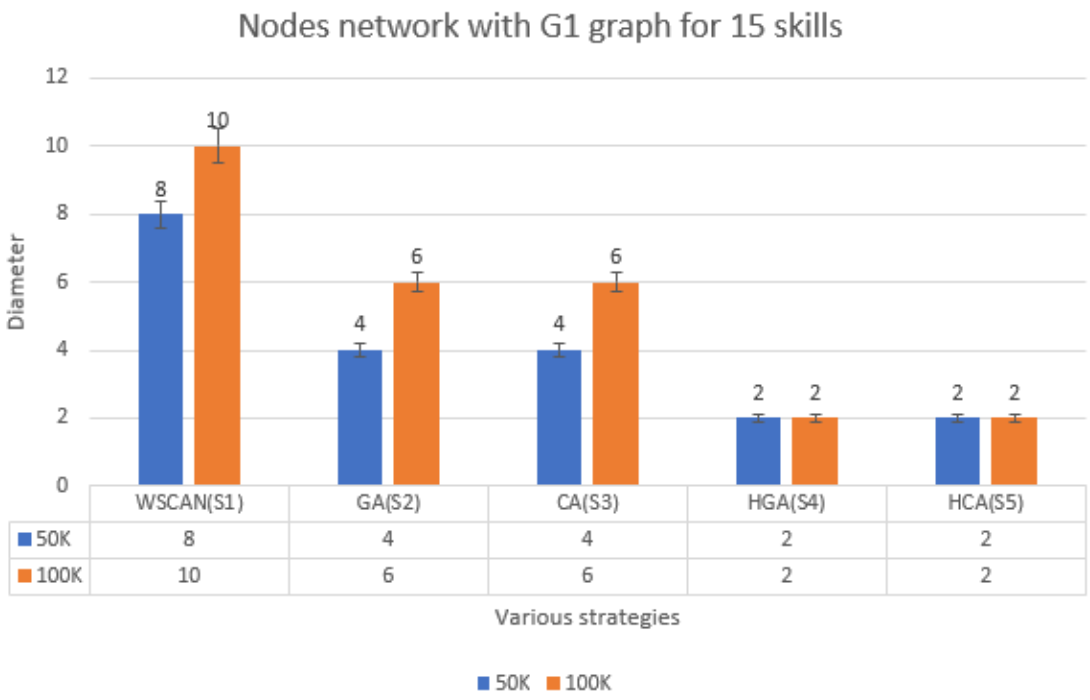


Figure 5.7: communication cost comparison with diameter for S1, S2, S3, S4 and S5

5.2.9 Effect of pool of experts (PoE) on results

Initially, we have complete node's network of size 50K (C1,C3,C5) and 100K (C2,C4,C6). We do not use complete node's network to find our team. We extract pool of experts (PoE) out of full sized network by removing unnecessary experts, we remove those experts which does not satisfy any skill from set of required skills. Pool of experts help us to find the team with minimum cost within less time. It means we are reducing our search space in order to find output with minimum cost in less time.

In this chapter, to prove the significance of results observed through experiments; we are using the t-test to prove it statistically, In order to prove any significant difference in observations, we collect sample population data as S_1 and S_2 in.

5.3 Performance measurement with average fitness score

We are comparing AFS, time taken (run-time) and Percentage Difference (i.e the absolute difference between the two values divided by the average of the two values [27]) for different heuristic with various set of required skills. Detailed discussion of each heuristic is given further in table 5.1.

Set of required skills	$AFS(S1)$	$AFS(S2)$	$AFS(S3)$	$AFS(S4)$	$AFS(S5)$
5 SKILLS	1543	1438	1105	486	467
10 SKILLS	7448	7169	6202	1120	1045
15 SKILLS	20567	19821	15029	7571	7434
20 SKILLS	36890	35766	35234	13584	12453
25 SKILLS	56765	55450	54218	18970	16785
30 SKILLS	91234	89717	76905	33296	31245

Table 5.1: Average Fitness Score comparison for S1, S2, S3, S4, and S5 using semantically weighted graph with 100K nodes network

5.3.1 Empirical analysis-Average Fitness Score (AFS)- S2 (GA) and S3 (CA)

To analyze based on these two quality measures, We compared all knowledge bases strategies with each other. The comparative analysis is shown below in table below.

As we can see in table 5.1, we are comparing S2 and S3 based on Average Fitness Score (AFS). The fitness function we defined for hybrid heuristic associated with a positive number for each individual. The best-suited individual should have a value

closer to zero as per defined by the fitness function. It means the individual with less value for AFS is fitter than the individual with high value at AFS. We run experiments for the various set of required skills, and we found that CA(S3) is fitter than GA(S2).

5.3.2 Average Fitness Score (AFS)- S4 (HGA) and S5 (HCA)

As we can see results in table 5.1, S4 and S5 based on Average Fitness Score (AFS). We run experiments for the various set of required skills, and we found that HCA(S5) is fitter than HGA(S4).

- **Percentage difference in Average Fitness** 3.90 percent improvement HGA to HCA.

5.3.3 Empirical analysis-Average Fitness Score (AFS)- S2 (GA) and S4 (HGA)

As we can see in a table 5.1, we are comparing S2 and S4 based on Average Fitness Score (AFS). We run experiments for the various set of required skills, and we found that HGA(S4) is fitter than GA(S2).

- **Percentage difference in Average Fitness** 66.20 percent improvement GA to HGA.

5.3.4 Empirical analysis-Average Fitness Score (AFS)- S3 (CA) and S5 (HCA)

As we can see in a table 5.1, we are comparing S5 and S3 based on Average Fitness Score (AFS). We run experiments for the various set of required skills, and we found that HCA(S5) is fitter than CA(S3).

- **Percentage difference in Average Fitness** 57.73 percent improvement CA to HCA.

5.3.5 Empirical analysis-Average Fitness Score (AFS)- S1 and S5

As we can see in a figure table 5.1, we are comparing non-knowledge based approach S1 with a knowledge-based approach S5. We are comparing with hybrid S5 because S5 has less value for AFS; which is good as per defined function. We run experiments for the various set of required skills, and we found that HCA(S5) is more fitter than HGA(S1) in terms of AFS.

- **Percentage difference in Average Fitness** 69.00 percent improvement WSCAN-TFP to HCA

5.3.6 Percentage difference- S1, S2, S3, S4 and S5

In figure 5.9, we calculated the percentage dierence between each of two methods (i.e., the absolute dierence between the two values divided by the average of the

two values)[26]. Percentage difference shows improvement in fitness of the solution between each of two strategies that we compared to each other for our experiments. In our case study, improvement in fitness of the solution means low fitness score on fitness function while we evaluate each individual solution.

Definition 36. *Percentage dierence: The absolute dierence between the two values divided by the average of the two values [26].*

$$\frac{N_1 - N_2}{\frac{(N_1+N_2)}{2}} \times 100 \tag{5.1}$$

- N_1 = Average fitness score from the first method
- N_2 = Average fitness score from the second method

AFS Percentage Difference	S_1	S_2	S_3	S_4	S_5
S_5	69.00%	67.52%	57.73%	3.90%	
S_4	68.50%	66.20%	56.01%		
S_3	28.38%	23.15%			
S_2	6.80%				

Figure 5.8: Percentage difference with AFS for S1, S2, S3, S4, and S5

5.3.7 Statistical analysis-AFS Comparison GA(S2)-HGA(S4)

To compare results from S2 and S4, we are defining the null hypothesis (H_0) and alternative thesis (H_a).

(H_0) = There is no statistically significant difference in average fitness score between GA (Sample size (S_1)=12) and HGA (Sample 2 (S_2)= 12).

(H_a) = There is statistically significant difference in average fitness score between GA (Sample size (S_1)=12) and HGA (Sample 2 (S_2)= 12).

Set of required skills	$\bar{x_1}$	$\bar{x_2}$	σ_1	σ_2	n_1	n_2	α	t	t_c
10 SKILLS	169.62	26.58	1.082	1.167	12	12	0.050	2.0288	1.72
15 SKILLS	455.62	202.87	1.192	1.110	12	12	0.025	4.3432	2.08
20 SKILLS	565.75	311.5	0.829	0.946	12	12	0.025	5.6336	2.08

Table 5.2: Showing data related to S2 and S4 for t-Test

- $\bar{x_1}$ = Sample mean (S_1)
- $\bar{x_2}$ = Sample mean (S_2)
- σ_1 = Standard deviation (S_1)
- σ_2 = Standard deviation (S_2)
- n_1 = Number of elements (S_1)
- n_2 = Number of elements (S_2)
- α = Significance level

- t = t- value
- t_c = Critical value

In chapter V, we are discussing the results we found after conducting various experiments mentioned in chapter VI.

To conduct the t-test, we collect sample data from HGA (Average Fitness Score) and GA (Average Fitness Score) in Case 1, Case 2 and Case 3. We are creating two independent sample from data observed, S_1 is HGA-AFS and S_2 is GA-AFS.

- **Case 1: 10 Skills**

  The t-value is 2.02882, and it is greater than the critical value $t_c = 1.72$ at significance level. So, the rejection region for this t-test is $R = \{t : t > 1.72\}$. Therefore, we reject the null hypothesis and accept the alternative hypothesis. There is a statistically significant difference between the two samples collected with AFS.

- **Case 2: 15 Skills**

  The t-value is 4.3432, and it is greater than the critical value at the significance level. So, the rejection region for this t-test is $R = \{t : t > 2.08\}$. Therefore, we reject the null hypothesis and accept the alternative hypothesis. There is a statistically significant difference between the two samples collected with AFS.

- **Case 3: 20 Skills**

  The t-value is 5.6336, and it is greater than the critical value at the significance level. So, the rejection region for this t-test is $R = \{t : t > 2.08\}$. Therefore, we

reject the null hypothesis and accept the alternative hypothesis. There is the statistically significant difference between two samples collected with AFS.

5.3.8 Statistical analysis-AFS Comparison CA(S3)-HCA(S5)

To compare results from S3 and S5, we are defining the null hypothesis (H_0) and alternative thesis (H_a).

(H_0) = There is no statistically significant difference in average fitness score between CA (Sample size (S_1)=14) and HCA (Sample 2 (S_2)= 14).

(H_a) = There is statistically significant difference in average fitness score between CA (Sample size (S_1)=14) and HCA (Sample 2 (S_2)= 14).

Set of required skills	$\bar{x_1}$	$\bar{x_2}$	σ_1	σ_2	n_1	n_2	α	t	t_c
10 SKILLS	134.642	21.785	3.494	2.284	14	14	0.05	3.9529	1.79
15 SKILLS	407.571	201.005	5.2288	3.1378	14	14	0.025	5.2189	2.20
20 SKILLS	514.892	305.071	5.6233	3.5129	14	14	0.025	7.3401	2.20

Table 5.3: Showing data related to S3 and S5 for t-Test

To conduct the t-test, we collect sample data from HCA (Average Fitness Score) and CA (Average Fitness Score) in Case 1, Case 2 and Case 3. We are creating two independent sample from data observed, S_1 is HCA-AFS and S_2 is CA-AFS.

- **Case 1: 10 Skills**

The t-value is 3.9529, and it is greater than the critical value at the significance level. Therefore, we reject the null hypothesis and accept the alternative hypothesis. There is a statistically significant difference between two samples collected with AFS.

- **Case 2: 15 Skills**

  The t-value is 5.2189, and it is greater than the critical value at the significance level. Therefore, we reject the null hypothesis and accept the alternative hypothesis. There is a statistically significant difference between two samples collected with AFS.

- **Case 3: 20 Skills**

  The t-value is 7.3401, and it is greater than the critical value at the significance level. Therefore, we reject the null hypothesis and accept the alternative hypothesis. There is a statistically significant difference between two samples collected with AFS.

5.4 Performance measurement with average processing time- S1, S2, S3, S4, and S5

To solve team formation problem in a social network researchers used various methods. Exact algorithm was implemented in [42] and [41] to find a solution for this problem. But author conducted experiments for small set of required skills that is 3 skills and 5 skills in figure 2.8. Exact algorithm gives exact solution but this algorithm takes months beyond this size of required skills to run and produce solution. This search

is exponential that means time-taken to find the team suddenly increase with small increase in set of required skills. In figure 5.9 we are comparing average processing time for all five strategies. Later in next section 5.5, we are conducting regression analysis of time-taken in milliseconds on Y-axis and set of required skills on X-axis.

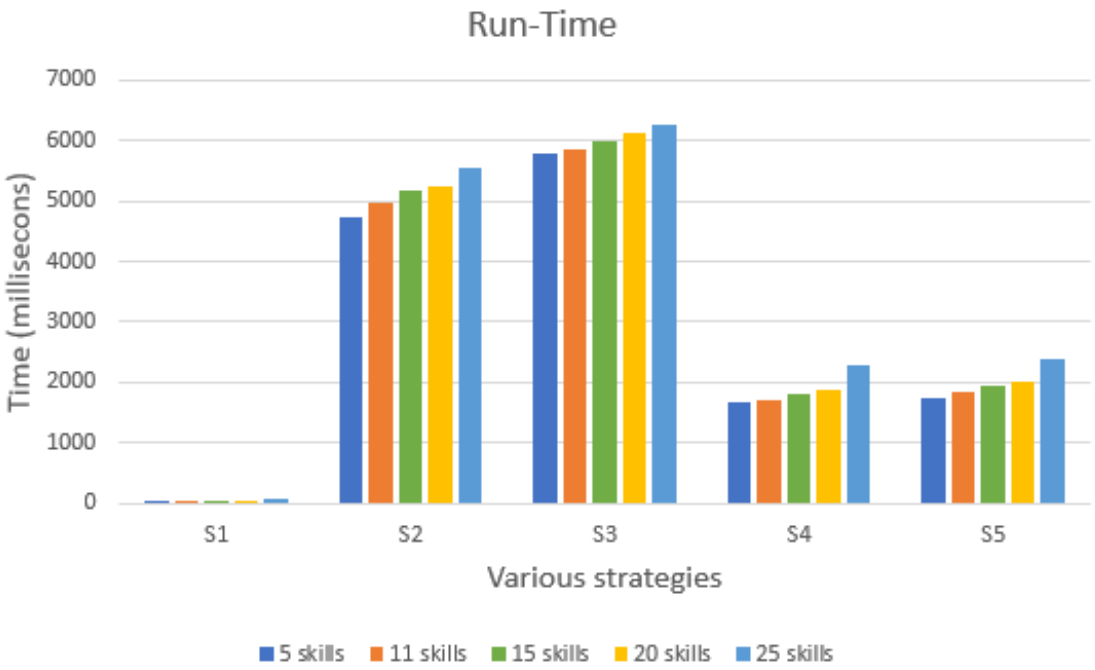


Figure 5.9: Shows time-taken (run-time) comparison for S1, S2, S3, S4, and S5 for different set of required skills

Set of required skills	S1 Time	S2 Time	S3 Time	S4 Time	S5 Time
5 SKILLS	30	4976	5869	1578	1667
7 SKILLS	32	5077	5922	1611	1742
10 SKILLS	34	5291	5997	1631	1774
12 SKILLS	35	5365	6133	1734	1954
15 SKILLS	37	5615	6399	2057	2289

Table 5.4: Time-taken in milliseconds comparison for S1, S2, S3, S4, and S5

The figure 5.9 above shows average time-taken by each heuristic to find a set of

required skills. As we can see, wscan-tfp is taking least time but when we compare average fitnesses score based on table 5.3 and average fitness based on average fitness score (less AFS value is better); we find that S1 gives lower quality compared to another knowledge-based heuristic. However in figure 5.9, when we compare all four knowledge-based heuristic S2, S3, S4, and S5; we find that S4 and S5 take less time comparative to S2 and S3. Moreover, it gives high performance **??** and high fitness (based on low AFS value) in table 5.3.

5.5 Regression Analysis

Regression analysis helps us to estimate relationship among the variables.

5.5.1 Exponential function

In mathematics, exponential function can be defined as a an independent variable x is exponent to a constant c and c is base.

Definition 37. *In mathematics, exponential function can be defined as a an independent variable x is exponent to a constant c.*

$$y = f(x) = c^x \tag{5.2}$$

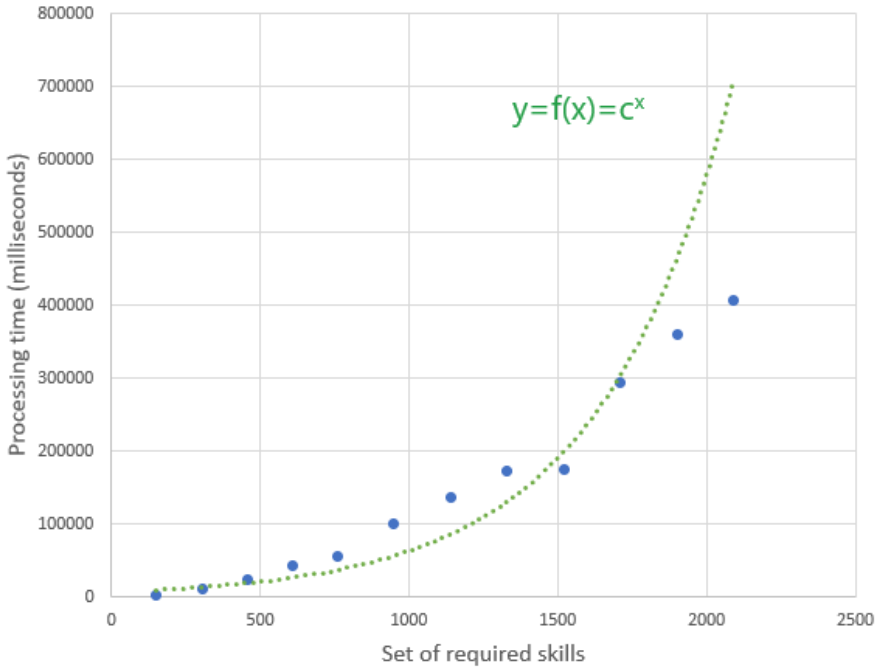


Figure 5.10: Regression analysis between processing time on Y-axis and set of required skills on X-axis for S5 with exponential function

In figure 5.10, we can see blue colored dotes show relationship among Y-axis and X-axis. We plotted processing time on Y-axis and set of required skills on X-axis. On the other hand green dotted line shows exponential function for the same graph which does not satisfy for our results for S5.

5.5.2 Power function

Definition 38. *In mathematics, power function can be defined as a an independent variable x is raised to a (constant variable) power c.*

$$y = f(x) = x^c \tag{5.3}$$

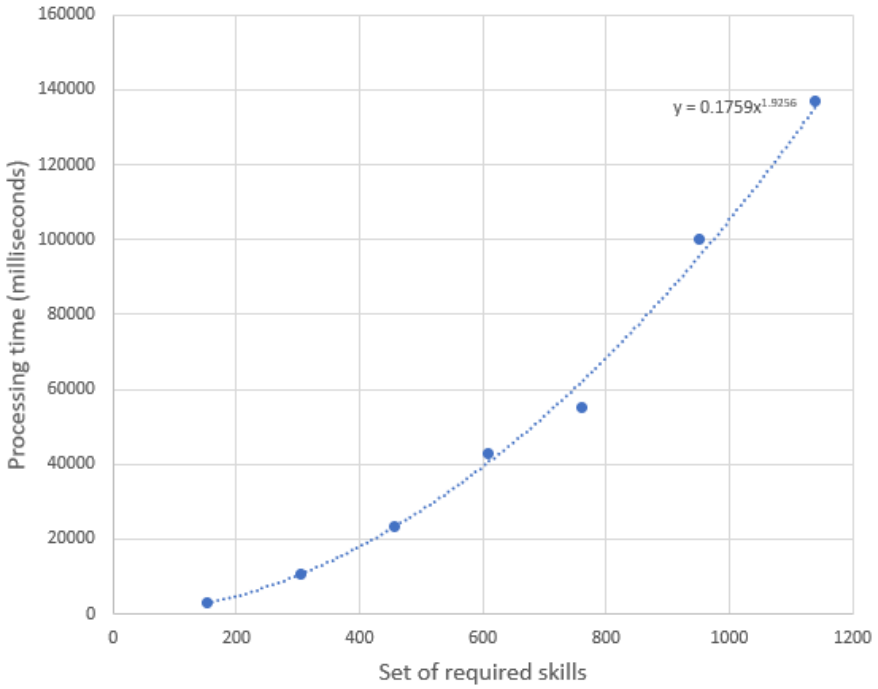


Figure 5.11: Regression analysis between processing time on Y-axis and set of required skills on X-axis for S5 (1050 skills)

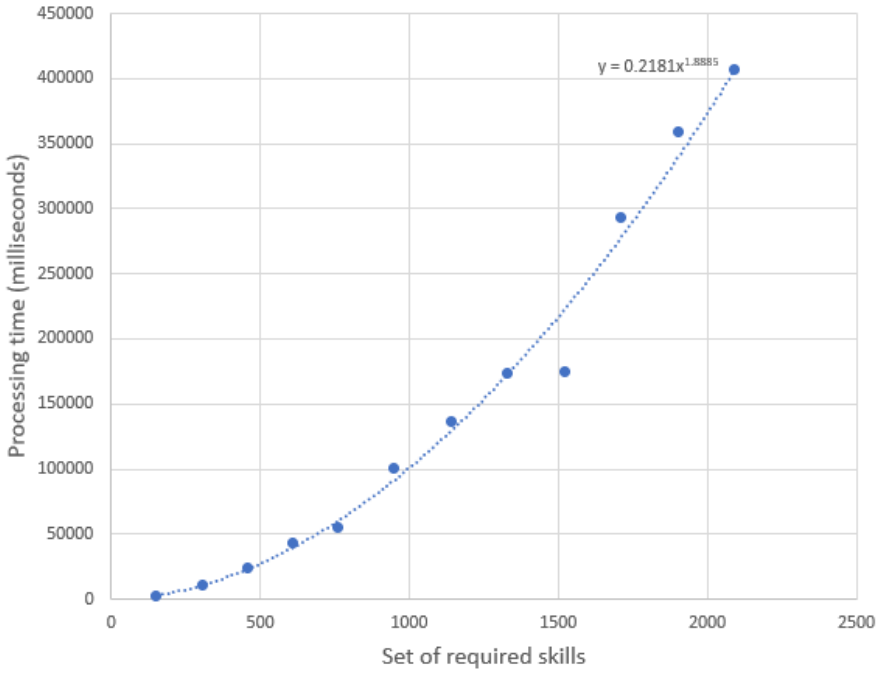


Figure 5.12: Regression analysis between processing time on Y-axis and set of required skills on X-axis for S5 (2050 skills)

In figure 5.11 and 5.12, we can see blue colored dotes show relationship among Y-axis and X-axis. We plotted processing time on Y-axis and set of required skills on X-axis. Whereas blue dotted line shows power function for the same graph which satisfy for our results for S5. In figure 5.11, we calculated time-taken to find a team for more than 1000 required skills and then we expanded our search for more than 2000 required skills in figure 5.12.

5.6 Limitations and Assumptions

5.6.1 Schema template construction

According to schema theorem by John Holland, we construct the schema template to increase chances to get desired results with GAs. So, construction of schema template highly depends upon the problem. We are using schema theorem to increase more chances to get desired results. In our experiments, we fix the set of skills satisfied by the core expert from set of required skills. So, processing time and communication cost highly depends upon construction of schema template. We can divide it into two cases. For example, Case I - Set of required skills R = (AI, DM, DB, ML, 3D, CG, LP, MT, PH ,CH), where R requires 10 skills to be satisfied by experts and form a team with those experts to complete a project. $[Core]_{e(i,j)}$ = (AI, DM, DB, ML, RB, AL, DAA). So, four skills are satisfied by core expert from R. Set of skills for Core expert ($[Core]_{e(i,j)}$) $\subseteq R$. Then, we use the schema theorem and construct schema template H. For those skills which matches with R,We fix core expert skills $e_{(i,j)} \in H$ in schema template H and mask it, we want to keep core expert because it is highly connect expert in the network based on structural similarity.We generate, population for rest of the remaining skills R'. Case II, R= (AI, DM, DB, ML, 3D, CG, LP, MT, PH ,CH), where R requires 10 skills same as case I. But, $[Core]_{e(i,j)}$ = (AI, DM, IS, IR, RB, AL, DAA). So, two skills are satisfied by core expert from R. So, case I shows

less communication cost comparative to case II. So in best case scenario, all skills are satisfied by the core expert. On the other hand in worse case scenario, R' are found with the outliers which have least structural similarity with the core expert. Case I takes less processing time than Case II.

5.6.2 Parameter used for generating nodes network

we assumed that sum of distance is only parameter for the nodes network we generated from DBLP. We are not considering any other parameters or/and combination of two or more parameters. Based on sum of distance, we are calculating communication cost and average fitness score. The proposed two strategies can be tested with other parameters or combination of parameters in future.

5.6.3 Evolutionary methods

There are abundant methods and approaches available in evolutionary computation to solve optimization problems such as team formation problem. We only tested and used handful techniques such as GAs, CAs and Schema theorem and proposed two hybrids based on these concepts with non-knowledge based clustering methods such as WSCAN that uses structural similarity concept. In future work proposed hybrids can be tested and improved for better results.

Chapter 6

Conclusion and Future Work

In this thesis, we experimented to solve the problem of finding the team of experts in a social network that covers the set of project specific skills with a minimum communication cost among team members with collective expertise.

6.1 Non-knowledge-based approach

To test on a non-knowledge based approach, we used a clustering approach based on structural similarity to test for TFP. We published WSCAN-TFP in [41], but in [41] paper we only compared these results with two evolutionary algorithms counterparts.

- Strategy 1 (S1):

  We implemented WSCAN-TFP to solve TFP, and we our findings indicate that WSCAN-TFP algorithm in strategy 1 (S1) works faster than the evolutionary counterparts, but it shows lower fitness (high AFS) compared to Strategy 2 (S2), Strategy 3 (S3), Strategy 4 (S4) and Strategy 5 (S5). Communication cost is higher than S2, S3, S4 and S5.

6.2 Knowledge-based Approach

We are proposing a new knowledge-based more efficient approach comparative to existing knowledge-based approaches for the same problem. We found the best possible case and worst possible case scenario. In the best case scenario, all skills from a set of project specific skills found within core expert and in worst case scenario, only one skill from Set of required skills belong to core expert and rest of the skills with other experts. But even in the worst case scenario for this algorithm, it is giving better results mostly comparative to other knowledge-based algorithms. Algorithm results are based on the construction of schema template. Hence, schema plays an important role in improved results of the Hybrid Genetic algorithm (HGA) and Hybrid Cultural Algorithm (HCA).

- Strategy 2 (S2): We implemented Genetic Algorithm to solve TFP, and we found that Genetic Algorithm in strategy 2 (S2) works slower than the WSCAN-TFP algorithm, but it shows better fitness (low AFS) compared to strategy 1 (S1). However, when we compared its average fitness with Strategy 3 (S3), Strategy 4 (S4) and Strategy 5 (S5), it shows high average fitness score comparatively, means less fitness than S3, S4, and S5. Communication cost is higher than S3, S4 and S5 but less than S1.

- Strategy 3 (S3): We implemented Cultural Algorithm to solve TFP, and we found that Cultural Algorithm in strategy 3 (S3) works slower than the WSCAN-TFP algorithm, but it shows better fitness (low AFS) compared to strategy 1 (S1) and strategy 2 (S2). However, when we compared its average fitness with

Strategy 4 (S4) and Strategy 5 (S5), it shows a higher average fitness score, it means lower fitness. Communication cost is higher than S4 and S5 but less than S1 and S2.

- Strategy 4 (S4): We implemented Hybrid Genetic Algorithm to solve TFP, and we found that Hybrid Genetic Algorithm in strategy 4 (S4) works slower than the WSCAN-TFP algorithm, but it shows better fitness (low AFS) compared to strategy 1 (S1), strategy 2 (S2) and Strategy 3 (S3). However, when we compared its speed with Strategy 5 (S5), it takes less time comparative to S5. But its shows lower average fitness than S5. Communication cost is higher than S5 but less than S1, S2, and S3.

- Strategy 5 (S5): We implemented Hybrid Cultural Algorithm to solve TFP, and we found that Hybrid Cultural Algorithm in strategy 5 (S5) works slower than the WSCAN-TFP algorithm but it shows better average fitness (low AFS) compared to strategy 1 (S1), strategy 2 (S2), Strategy 3 (S3) and Strategy 4 (S4). But it takes more time than S4 to find the individuals/solution. Communication cost is less than S1, S2, S3 and S4.

6.3 Future Work

Team Formation Problem (TFP) can be seen from various angles. In our experiment, we are only considering communication cost as a parameter to find teams from a Social Network (SN). However, Workload can be a different parameter to be considered with this algorithm in the future. Workload over each expert can affect the output of the team for a specific project. Moreover, salary determination, the efficiency of

a team and geographical distance can be seen as a future extension of this work. This experiment is done on the static environment. In addition to this, dynamic environment for TFP with communication cost as a parameter can also be seen as a future extension of this research work. Further, considering all parameters separately, a combination of different parameters can be considered together. Apart from this, strategy 4 (S4) and strategy 5 (S5) can be tested on other optimization problem.

Bibliography

[1] Lada A Adamic, Jun Zhang, Eytan Bakshy, and Mark S Ackerman. Knowledge sharing and yahoo answers: everyone knows something. In *Proceedings of the 17th international conference on World Wide Web*, pages 665–674. ACM, 2008.

[2] Luis Agustin-blas, Sancho Salcedo-Sanz, Emilio Ortiz-Garcia, Antonio Portilla-Figueras, Angel Perez-Bellido, and Silvia Jimenez-Fernandez. Team formation based on group technology: a hybrid grouping genetic algorithm approach. *IEEE Engineering Management Review*, 1(40):30–43, 2012.

[3] Takuya Akiba, Y. Iwata, and Y. Yoshida. Fast Exact Shortest-Path Distance Queries on Large Networks by Pruned Landmark Labeling. In *SIGMOD*, pages 349–360, 2013.

[4] Aris Anagnostopoulos, Luca Becchetti, Carlos Castillo, Aristides Gionis, and Stefano Leonardi. Power in unity: forming teams in large-scale community systems. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 599–608. ACM, 2010.

[5] Aris Anagnostopoulos, Luca Becchetti, Carlos Castillo, Aristides Gionis, and Stefano Leonardi. Online team formation in social networks. In *Proceedings of the 21st international conference on World Wide Web*, pages 839–848. ACM, 2012.

[6] Zhamri Che Ani, Azman Yasin, Mohd Zabidin Husin, and Zauridah Abdul Hamid. A method for group formation using genetic algorithm. *International Journal on Computer Science and Engineering*, 2(9):3060–3064, 2010.

[7] Gaganmeet Kaur Awal and KK Bharadwaj. Team formation in social networks based on collective intelligence–an evolutionary approach. *Applied Intelligence*, 41(2):627–648, 2014.

[8] Marko Balabanović and Yoav Shoham. Fab: content-based, collaborative recommendation. *Communications of the ACM*, 40(3):66–72, 1997.

[9] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *science*, 286(5439):509–512, 1999.

[10] Punam Bedi and Chhavi Sharma. Community detection in social networks. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 6(3):115–135, 2016.

[11] Lashon Bernard Booker, David E Goldberg, and John Henry Holland. Classifier systems and genetic algorithms. 1989.

[12] Stephen P Boyd and Lieven Vandenberghe. Convex optimization (pdf). *Np: Cambridge UP*, 2004.

[13] Anton Chertov. Extension of graph clustering algorithms based on scan method in order to target weighted graphs. 2012.

[14] Anton Chertov, Ziad Kobti, and Scott D Goodwin. Weighted scan for modeling cooperative group role dynamics. In *Computational Intelligence and Games (CIG), 2010 IEEE Symposium on*, pages 17–22. IEEE, 2010.

[15] Chan-Jin Chung. Knowledge-based approaches to self-adaptation in cultural algorithms. 1997.

[16] E. Cohen, E. Halperin, H. Kaplan, and U. Zwick. Reachability and Distance Queries via 2-hop Labels. In *SODA*, pages 937–946, 2002.

[17] White David. An overview of schema theory. *arXiv preprint arXiv:1401.2651*, 2014.

[18] Christoph Dorn and Schahram Dustdar. Composing near-optimal expert teams: a trade-off between skills and connectivity. In *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, pages 472–489. Springer, 2010.

[19] Abdulrahman M El-Sayed, Peter Scarborough, Lars Seemann, and Sandro Galea. Social network analysis and agent-based modeling in social epidemiology. *Epidemiologic Perspectives & Innovations*, 9(1):1, 2012.

[20] Alireza Farasat and Alexander G Nikolaev. Social structure optimization in team formation. *Computers & Operations Research*, 74:127–142, 2016.

[21] Amita Gajewar and Atish Das Sarma. Multi-skill collaborative teams based on densest subgraphs. In *Proceedings of the 2012 SIAM International Conference on Data Mining*, pages 165–176. SIAM, 2012.

[22] Vicenç Gómez, Andreas Kaltenbrunner, and Vicente López. Statistical analysis of the social network and discussion threads in slashdot. In *Proceedings of the 17th international conference on World Wide Web*, pages 645–654. ACM, 2008.

[23] Yuqiang Han, Yao Wan, Liang Chen, Guandong Xu, and Jian Wu. Exploiting geographical location for team formation in social coding sites. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 499–510. Springer, 2017.

[24] John Holland. Adaptation in natural and artificial systems. 1 ?edigao. *Ann Arbor, USA: The University of Michigan Press*, 1975.

[25] John Henry Holland. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press, 1992.

[26] Mehdi Kargar and Aijun An. Discovering top-k teams of experts with/without a leader in social networks. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 985–994. ACM, 2011.

[27] Mehdi Kargar, Aijun An, and Morteza Zihayat. Efficient bi-objective team formation in social networks. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 483–498. Springer, 2012.

[28] Mehdi Kargar, Morteza Zihayat, and Aijun An. Finding affordable and collaborative teams from a network of experts. In *Proceedings of the 2013 SIAM International Conference on Data Mining*, pages 587–595. SIAM, 2013.

[29] Miray Kas, Kathleen M Carley, and L Richard Carley. Who was where, when? spatiotemporal analysis of researcher mobility in nuclear science. In *Data Engineering Workshops (ICDEW), 2012 IEEE 28th International Conference on*, pages 347–350. IEEE, 2012.

[30] Jinho Kim, Kyuseok Shim, Longbing Cao, Jae-Gil Lee, Xuemin Lin, and Yang-Sae Moon. *Advances in Knowledge Discovery and Data Mining: 21st Pacific-Asia Conference, PAKDD 2017, Jeju, South Korea, May 23-26, 2017, Proceedings*, volume 10234. Springer, 2017.

[31] Theodoros Lappas, Kun Liu, and Evimaria Terzi. Finding a team of experts in social networks. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 467–476. ACM, 2009.

[32] Cheng-Te Li, Man-Kwan Shan, and Shou-De Lin. On team formation with expertise query in collaborative social networks. *Knowledge and Information Systems*, 42(2):441–463, 2015.

[33] Ying-Hong Liao and Chuen-Tsai Sun. An educational genetic algorithms learning tool. *IEEE transactions on Education*, 44(2):20–pp, 2001.

[34] Anirban Majumder, Samik Datta, and KVM Naidu. Capacitated team formation problem on social networks. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1005–1013. ACM, 2012.

[35] Pooya Moradian Zadeh. Social network analysis using cultural algorithms and its variants. 2017.

[36] Panth Parikh. Knowledge migration strategies for optimization of multi-population cultural algorithm. 2017.

[37] Syama Sundar Rangapuram, Thomas Bühler, and Matthias Hein. Towards realistic team formation in social networks based on densest subgraphs. In *Proceedings of the 22nd international conference on World Wide Web*, pages 1077–1088. ACM, 2013.

[38] Robert G Reynolds. An introduction to cultural algorithms. In *Proceedings of the third annual conference on evolutionary programming*, pages 131–139. World Scientific, 1994.

[39] Robert G Reynolds and Bin Peng. Cultural algorithms: modeling of how cultures learn to solve problems. In *Tools with Artificial Intelligence, 2004. ICTAI 2004. 16th IEEE International Conference on*, pages 166–172. IEEE, 2004.

[40] John Scott. *Social network analysis*. Sage, 2017.

[41] Kalyani Selvarajah, Amangel Bhullar, Ziad Kobti, and Mehdi Kargar. Wscan-tfp: Weighted scan clustering algorithm for team formation problem in social network. In *FLAIRS Conference*, pages 209–212, 2018.

[42] Kalyani Selvarajah, Pooya Moradian Zadeh, Mehdi Kargar, and Ziad Kobti. A knowledge-based computational algorithm for discovering a team of experts in social networks. 2017.

[43] Hiroaki Shiokawa, Yasuhiro Fujiwara, and Makoto Onizuka. Scan++: efficient algorithm for finding clusters, hubs and outliers on large-scale graphs. *Proceedings of the VLDB Endowment*, 8(11):1178–1189, 2015.

[44] Thanh N Tran, Klaudia Drab, and Michal Daszykowski. Revised dbscan algorithm to cluster data with dense adjacent clusters. *Chemometrics and Intelligent Laboratory Systems*, 120:92–96, 2013.

[45] Santosh Upadhyayula. Dominance in multi-population cultural algorithms. 2015.

[46] Renee C Van der Hulst. Introduction to social network analysis (sna) as an investigative tool. *Trends in Organized Crime*, 12(2):101–121, 2009.

[47] Peng Wang, BaoWen Xu, YuRong Wu, and XiaoYu Zhou. Link prediction in social networks: the state-of-the-art. *Science China Information Sciences*, 58(1):1–38, 2015.

[48] Stanley Wasserman and Katherine Faust. *Social network analysis: Methods and applications*, volume 8. Cambridge university press, 1994.

[49] Peter A Whigham. A schema theorem for context-free grammars. In *Evolutionary Computation, 1995., IEEE International Conference on*, volume 1, page 178. IEEE, 1995.

[50] Hyeongon Wi, Seungjin Oh, Jungtae Mun, and Mooyoung Jung. A team formation model based on knowledge and collaboration. *IEEE Engineering Management Review*, 1(40):44–57, 2012.

[51] Xiaowei Xu, Nurcan Yuruk, Zhidan Feng, and Thomas AJ Schweiger. Scan: a structural clustering algorithm for networks. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 824–833. ACM, 2007.

[52] Morteza Zihayat, Aijun An, Lukasz Golab, Mehdi Kargar, and Jaroslaw Szlichta. Authority-based team discovery in social networks. *arXiv preprint arXiv:1611.02992*, 2016.

Appendix A

// All vertices are unclassified
For each unclassified vertex $v \in V$ do
// step 1. Check whether v is a core;
if $Core_{\epsilon,\mu}(v)$ then
// step 2.1 if v is core, a new cluster is expanded;
Generate new clusterID;
insert all $x \in N_{\epsilon}(v)$ into queue Q;
While $Q \neq 0$ do
$y =$ first vertex in Q;
$R = x \in V | DirREACH_{\epsilon,\mu}(x, y)$;
For each $x \in R$
if x is unclassified or non-member then
assign current clusterID to x;
if x is unclassified then
insert x into queue Q;
remove y from Q;
Else
// step 2.2 if v is not a core, it is labeled as non-member
Label v as a non-member;
End for
// step 3. Further classifies non-members
For each non-member vertex v do
if $\exists x, y \in \tau(v)(x.clusterID \neq y.clusterID)$ then
Label v as a hub
Else
Label v as outlier;
End for
End SCAN

Algorithm 5: SCAN Algorithm by author in [51]

Vita Auctoris

NAME:	Amangel Bhullar
PLACE OF BIRTH:	Punjab,India
EDUCATION:	Bachelor of Technology in Computer Science, Punjab Technical University, Punjab, India, 2012.
	Master of Science in Computer Science, University of Windsor, Windsor, Ontario, Canada, 2018.