A Chemistry-Oriented Cost-Effective Expert Team Formation in Social Networks

by

Yashar Najaflou

A thesis
submitted to the Victoria University of Wellington
in fulfilment of the
requirements for the degree of
Master of Science
in Computer Science.

Victoria University of Wellington
2017

Abstract

The growth of social networks in modern information systems has enabled the collaboration of experts at an unprecedented scale. Given a social network and a task consisting of a set of required skills, Team Formation (TF) aims at finding a team of experts who can cover the required skills and can communicate in an effective manner. However, this definition has been interpreted as the problem of finding teams with minimum communication cost which neglects two aspect of team formation in real life. The first is that in reality experts are multi-skilled, hence communication cost cannot be a fixed value and should vary according to the channels employed. The second ignored aspect is disregarding teams with high expertise level who can still satisfy the required communication level.

To tackle above mentioned issues, I introduce a dynamic form of communication for multi-facet relationships and use it to devise a novel approach called Chemistry Oriented Team Formation (ChemoTF) based on two new metrics; Chemistry Level and Expertise Level. Chemistry Level measures scale of communication required by the task and Expertise Level measures the overall expertise among potential teams filtered by Chemistry Level. Moreover, I adopt a personnel cost metric to filter costly teams. The experimental results on the corpus compiled for this purpose suggests that ChemoTF returns communicative and cost-effective teams with the highest expertise level compared to state-of-the-art algorithms. The corpus itself is a valuable output which contains comprehensive scholarly information in the field of computer science.

Keywords: Team Formation, Expert, Social Network, Chemistry Level

Acknowledgements

Foremost, I would like to thank my supervisor, Dr Kris Bubendorfer, for the guidance, encouragement, and advice that he has provided throughout my time as his student. I have been lucky to have a supervisor who cared about my work, who responded to my questions promptly, and who acted in such a way that I felt close to him. I believe in order to describe my supervisor, it is fair to say that he has been the best experience in my entire academic life and I am glad to have been part of his too.

I must also express my gratitude to Neda, my wife, for her continued support and encouragement. She was extremely patient to tolerate me during ups and downs of my research. Completing this work would have not been possible without her efforts to compensate my continual negligence of our family and her willingness to proof read countless pages of meaningless mathematics and formulas. I am indebted to you for your help and kindness.

Finally, my special thanks goes to the rest of my family for their motivation and support, and the staff members and postgraduate students in the School of Engineering and Computer Science of Victoria University of Wellington for their patience and encouragement. I particularly appreciate my school's patience and support when I broke the network which slowed things down for the students and staff. Thanks for believing in my research and providing this opportunity for me to beat the beast.

List of Figures

List of Tables

Contents

"The isolated man does not develop any intellectual power. It is necessary for him to be immersed in an environment of other men."

Alan Turing quoted in [69]

Chapter 1

Introduction

Teamwork is probably one of the oldest and most important concepts in human society. Over the centuries, human beings far superior than other species have learnt to work as teams rather than individuals to empower their way of working, foster camaraderie, promote equity, and achieve the impossible. Nowadays we have no doubt that working within teams is an indisputable component of our success, particularly in modern societies where everyone and everything is connected thanks to the new technology. The advent of online communities and social networks has unfettered the idea of a cooperative world where events occur rapidly in collaboration of people within groups.

However, such an expeditious transformation has transformed the intrinsic nature of forming teams. Having access to a vast number of people in almost no time has expanded the expectations of teams to an extent that team formation has become an important research topic in numerous fields. This thesis contributes to the concept of team formation in the field of computer science by proposing a novel approach to overcome the challenges encountered while finding team of experts that can perform tasks with the best possible outcome. This research is an effort to bring to light the missing piece of the old concept of team formation in social networks and open up new doors for future research.

1.1 Team Formation and Limitations

Teamwork is often considered an important factor of success in projects. Working within teams is driven by human instinct to live and work within groups. In addition, the advent of social networks has encouraged individuals to contribute with each other more than ever. Given a social network and a task consisting of a set of required skills, Team Formation (TF) aims at finding a team of experts who not only satisfy the requirements of the given task, but also are able to communicate with one another in an effective manner [105]. However, choosing experts for the team, comparing the quality of different teams, and assessing the outcome, have various limitations. The heart of these limitations lies in definition of effective communication which leads to miscalculation in quantifying communication cost between experts.

TF has been often interpreted as the problem of finding teams with minimum cost of communication. Considering effective communication as communication with minimum cost is debatable, subjective and far from reality. Experts in reality have multiple skills and their relationships are multi-faceted. The form and strength of their relationship depends on the topic in which the communication between experts takes place. This means that relationships encompass various communication channels. Experts chose an appropriate channel to communicate depending on the skills they employ during their experience. Let's describe this in an example:

Imagine a relationship between two experts called Alice and Colin. Alice has expertise on "C++" whereas Colin posses two skills including "Java" and "Photography". If Alice communicates to Colin on the topic of "Programming", they will have a strong connection because their corresponding skills "C++" and "Java" are interconnected in this topic. However if the topic is "Graphics", they have less common ground thus they will have a much weaker communication. This example suggests that the

cost of communication between two experts is not independent of skills and cannot be concluded to a fixed value. It also suggests that the concept of relationship and communication should be distinguished. In order to match these concepts with reality, a dynamic form of communication cost is required which is bound to the skills employed in a relationship.

The lack of a proper understanding of communication cost has an unfavourable effect on the quality of the final teams as well. The minimum communication requirement is so strict that leaves no room for other aspects of a successful teams such as the overall expertise level of the team to be considered. In other words, filtering teams based on the lowest communication cost disregards teams with an overall higher expertise level which can still satisfy required communication level. If the required communication cost for a task is estimated, it can be used as a threshold to filter teams that elude the required level of communication for a given task. Considering relationships as multinomial as described earlier, this threshold should be subject to change according to skills employed.

In order to provide a better picture of the limitations around TF, the following section exemplifies a scenario where there exists a social network of experts and the aim is to find the best possible team to fulfil a given task.

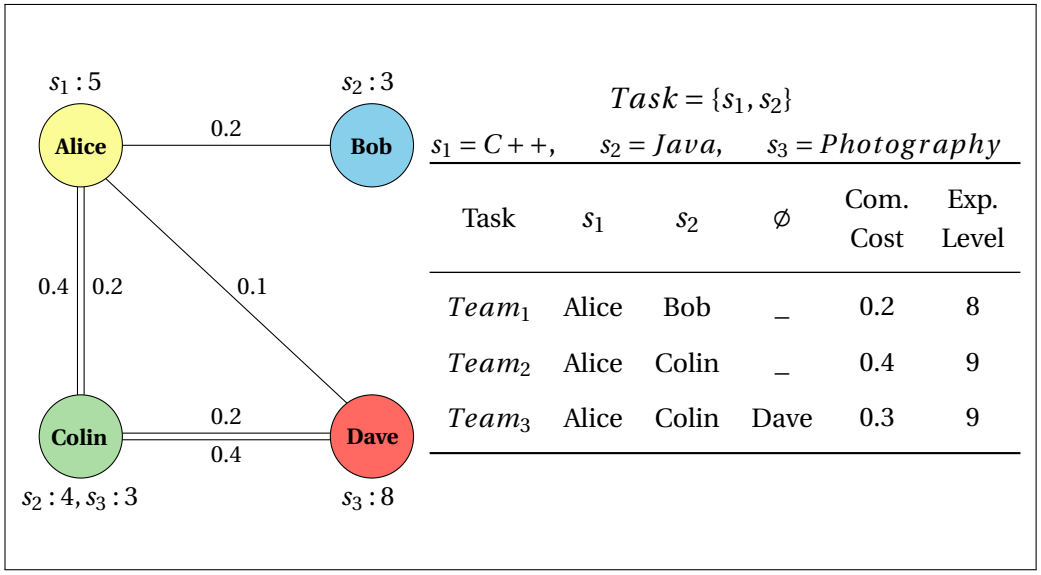
1.2 Motivational Example

Let's assume a network of experts in which individuals possess a set of skills and connected through their relationships. In this network, the relationships are multi-faceted, meaning that they are inclusive of all mutual experiences between experts. The communication cost required for each experience to take place determines the weights of the connections. Since some experts are multi-skilled, they can have multiple unique ex-

periences with one another. Each unique experience takes different set of skills to occur hence every experience is associated with a different communication cost value. Furthermore, each expert is associated with a number referring to their expertise level. Let's assume the expertise level of each expert represents the number of the years he or she has experience in particular skills.

An example of a social network of experts with the above mentioned attributes has been illustrated on left side of the Figure 1.1. In this example, Alice (yellow), Bob (blue), Colin (green), and Dave (red) construct a network according to their historic relationships. The weights on the edges represent communication costs. Since Colin has two skills, he has two sets of experiences with Alice and Dave, each with a different communication cost. This means if Colin would like to communicate with Alice using his s_2 skill, he uses the left channel which has been associated with 0.4 as the communication cost. However if Colin decides to communicate with Alice using his s_3 skill, he has to use the right channel with a communication cost of 0.2. The same scenario applies to the relationship between Colin and Dave. The expertise levels of experts have been demonstrated in front of the skills the expertise levels correspond to. For example, Alice is associated with a value of 5 in s_1.

Having an example network in hand, let's assume s_1 represents "C++", s_2 reflects "Java", and s_3 denotes "Photography" and define a task consisting of s_1 and s_2. We can now explore the network by walking through the connections in order to find teams of experts who can cover this task. An expert is selected as the starting point and then the path is walked to the other experts until all the required skills are covered. This means that the teams must be directly or indirectly connected. In the case of indirect connections, intermediate nodes are taken as part of the team. Since these intermediates might not cover any required skill, they are listed under $\emptyset$. The communication cost of the final team is the sum of communication cost of all the paths taken and the expertise level is the sum of the



$Task = \{s_1, s_2\}$

$s_1 = C++, \quad s_2 = Java, \quad s_3 = Photography$

Task	s_1	s_2	$\emptyset$	Com. Cost	Exp. Level
$Team_1$	Alice	Bob	_	0.2	8
$Team_2$	Alice	Colin	_	0.4	9
$Team_3$	Alice	Colin	Dave	0.3	9

Figure 1.1: An Example of expertise network in which teams are discovered to cover a task

expertise level of all the experts in the skill they cover.

As shown in the table on the right side of the Figure 1.1, three teams can be discovered to fulfil the task. $Team_1$ is formed by Alice and Bob for which communication cost is 0.2 and the sum of expertise level is 8. $Team_2$ is constructed by Alice and Colin given his s_2 skill with a communication cost of 0.4 and an expertise level of 9. The last team includes Alice and Colin which is similar to $Team_2$ but from Dave's path. This means that unlike the first two teams, $Team_3$ has Dave as intermediate. This team has a communication cost value less than the second team at 0.3 and an expertise level identical to the second team at 9.

Comparing the three teams, it can be identified that $team_1$ reflects the lowest communication cost among all the teams. In other words given the required skills, communication cost for Alice and Bob to perform the task is less than any other team across the network. From expertise level perspective though, $team_2$ and $team_3$ are the best teams as they reflect the highest expertise level among all three teams. This means that if the suc-

cess of a team is solely derived from the communication between team members, then $team_1$ is the only team to consider because it mirrors the least cost which leads to the hypothesis bellow:

Hypothesis 1.1. *Assuming that the success of a team can rely upon various parameters, the requirement of minimum communication cost can be relaxed in order to make room for other parameters such as the overall expertise of teams members.*

The argument above raises a fundamental question of "how to ensure that teams with higher communication cost can communicate effectively?". The answer to this question might lie in the nature of the skills. In the given example, "C++", "Java", and "Photography" were defined as skills. Given that "C++" and "Java" are both programming languages, they share a huge number of attributes with each other. Thus experts possessing these two skills have a common ground and can be expected to communicate well. Comparing the social chemistry between the above mentioned pair of skills to the chemistry between "C++" and "Photography", one can easily identify less common ground between the skills in the latter case hence less chemistry level between them in overall. This suggests that expecting experts possessing "C++" and "Photography" to communicate at the same level that experts possessing "C++" and "Java" do is misleading and leads to the hypothesis bellow:

Hypothesis 1.2. *Assuming that there is a meaningful and quantifiable chemistry between skills, a threshold can be configured according to the task reflecting the minimum communication cost required to perform the task. Teams with the same level of communication or higher than the chemistry between their skills can then be considered to communicate effectively.*

As an outcome of the discussion above and assuming that chemistry between s_1 and s_2 in the given example is calculated as 0.5, all three teams are considered to communicate effectively. This helps consider expertise

level of the teams as a secondary parameter based on the rationale that more expertise leads to quality outcomes. If this factor is considered, then $Team_1$ is ignored and $Team_2$ and $Team_3$ with expertise level of 9 become potential candidates. This raises another fundamental question which is "how to chose the one team among all the communicative teams with high expertise level?". The answer might be manifested in the economical aspect of teams and can be explained by the following hypothesis:

Hypothesis 1.3. *Assuming that teams with higher cost of experts lead to costly projects, teams that are associated with the least cost and satisfy the project budget are more likely to succeed.*

Given the hypothesis above, $Team_3$ is associated with more cost than all the other teams thus is expected to be less successful. Whatever the cost of Dave in this team is, he increases the overall cost of the team. This suggests that $Team_2$ is financially more justifiable compared to $Team_3$ thus it is chosen as the final team given the task and underling social network.

The given motivational example and highlighted hypothesis describe how various parameters can be employed to form teams for a given task. The rationale behind forming these teams requires a systematic mechanism in which all the fundamental questions raised in the given example are addressed. In the following section, I summarise these questions along with some other complementary objectives of this study.

1.3 Research Questions

There is no definitive answer to the question of what the best team is for a given task. In fact the problem of Team Formation in social networks has been proven to be *NP-hard* [86]. However, there are many more criteria involved while forming effective teams than choosing the team with minimum communication cost. This thesis covers a few number of pos-

sibilities that can be objectively explored in team formation. The main concern of this thesis is:

- *How to form cost-effective team of experts with the highest level of expertise possible that can cover the required skills in a given task and its members are able to communicate with each other effectively?*

More specifically, this study aims to provide objective answers to the following research questions in the context of computer science:

1. *How to estimate dynamic communication cost which is computed according to the communication channel employed in a relationship?*
2. *How to discover the required communication level for a given task according to the chemistry level between the skills it contains?*
3. *How to quantify the expertise level of a team of experts?*
4. *How to ensure the formed teams are cost-effective?*
5. *How to design a mechanism in which all the essential parameters of team formation are incorporated?*

1.4 Thesis Outline

The introductory chapter of this thesis is followed by a literature review chapter which takes a close look at team formation in its classic and modern era. An extensive survey is conducted to provide an understanding of the problem and proposed solutions. Furthermore, various factors employed in addressing team formation are identified and the existing research body are categorised accordingly.

Chapter 3 describes my proposed approach towards team formation called Chemistry Oriented Team Formation (ChemoTF) to addresses team

formation problem in social networks. First the related concepts and parameters are defined and the problem is formulated accordingly. Then an analytical model and an algorithm for ChemoTF are presented for the first time. In addition, the mechanism of ChemoTF to tackle the TF problem is explained.

In Chapter 4 I discuss how a corpus consisting of experts and their social ties is designed and compiled. This corpus which is called Comprehensive Scholarly Corpus (CompScholarCorp) contains unique scholarly information of publications such as keywords, abstracts and citations in the field of computer science. The process of design, data collection and sampling to compiled this corpus is thoroughly explained and examples of the records in this corpus are outlined.

CompScholarCorp is visualised in a series of illustrations, graphs and statistical tables in Chapter 5. This chapter eases the understanding of the concepts introduced in this thesis and identifies the possibilities that my corpus can be used for further research. First an overview of CompScholarCorp and its entities is given. Then the corpus is visualised from the various perspectives. In addition the networks derived from this corpus are discussed and visualised.

Chapter 6 describes how the corpus is employed as a testbed to model multi-faceted relationships between experts, calculate the parameters of ChemoTF, and construct a multidimensional network of experts. First the relationships between experts are modelled using a specific instance of probabilistic generative topic models called *Latent Dirichlet Allocation* (LDA) [22]. Then the constructed model is used to build areas of expertise and form the relationship between experts in each area. Finally the scale of contribution of each area of expertise in relationships is calculated using a machine learning inference technique called *Variational Inference* [75] and a multidimensional network is drawn accordingly. The model is evaluated using an approximation technique called *Perplexity* [22] to ensure that it reflects the reality as expected.

Having prepared all the required components of ChemoTF, Chapter 7 explains how the algorithm was implemented for the experiment. The designed and implemented platform covers ChemoTF along with three state-of-the-art TF algorithms to conduct comparative analysis. First the requirement analysis conducted to identify the core objective and mechanism of each algorithm is described. In addition, the requirements regarding the implementation techniques and technologies are identified. Then a database is designed to facilitate the process of providing necessary data from the corpus for the algorithm during the experiment. Finally I explain how the identified core objectives are implemented, verified and expanded to cover all the requirements with respect to data and functionalities.

In Chapter 8, experiments are conducted on the corpus and the results are analysed. First the environment where the experiment was conducted is discussed and the metrics of the analysis are outlined. Then the experimental results of my algorithm are analysed and the quality of the teams it forms is assessed. In addition, a comparative study is conducted to evaluate ChemoTF results against the three state-of-the-art algorithms. Finally a sensitivity analysis is conducted where the effect of parameters of ChemoTF is studied.

The final chapter concludes this thesis by summarising the findings, presenting the contributions, and explaining limitations of this study as well as the suggestions for future research.

> "Creating a new theory is rather like climbing a mountain, gaining new and wider views, discovering unexpected connections between our starting points and its rich environment."
>
> *Elbert Einstein* [49]

Chapter 2

Literature Review

There is a considerable amount of research on the Team Formation problem in various fields such as psychology, philosophy, sociology, management, mathematics, computer science, etc. What has made this problem so interesting is its practicality which is driven by human instinct to live and work within groups. In addition, the advent of social network as a new phenomenon has made the problem of forming teams even more appealing than ever before and has required researchers to devise more pragmatic approaches toward Team Formation. Therefore, the studies on the Team Formation problem fall into two categories namely Classic Team Formation prior to social network phenomenon and Team Formation in Social Network.

In this chapter an extensive review of the previous studies on Team Formation with respect to the above mentioned categories has been provided. First the existing studies in classic Team Formation era are discussed. Then various parameters and metrics employed in addressing team formation in social networks are identified and categorised and the existing research are outlined according to these categories. In addition, various applications of team formation in other fields are outlined and briefly discussed. Finally, the findings are summarised and the gaps are elaborated.

2.1 Classic Team Formation

Classic Team Formation is referred to as an era when researchers did not consider social ties among people in their approach to tackle team formation. The social ties in this era were in the form of traditional or small communities where the relationships between people were not transparent enough for researchers to conduct proper studies on. Most studies in this era were either theoretical or small scale such as small communities or organisational space. Hence the definition of Team Formation (TF) and the problems associated with it were interpreted according to the necessities or capacity of the research.

What makes TF different from a classic *Set Cover* problem [79] is the quality of communication between team members as a measure of a successful performance. A team in Set Cover problem is a set of individuals who have a finite number of skills. An optimum team in this problem contains "members who have the necessary skills to complete the task" [123]. TF has a fundamentally different approach toward a desirable team. Cohen and Bailey [36] define a team as:

> *"Collection of individuals who are independent in their tasks, who share responsibility for outcomes, who see themselves and are seen by others as an intact social entity embedded in one or more larger social systems, and who manage their relationships across organisational boundaries."*

Based on this definition of a team, Owen *et al.* [105] describe TF as:

> *"A result of the deliberate, strategic decisions of individuals who either self-select or assign others to a team with the purpose of satisfying individual and team objectives."*

Authors in this work inspected the effect of task-driven responsibilities within the team on the overall performance of the team in the organisational context. They found that a successful team is composed of

people who have skills to perform the task and a good level of interpersonal relationships. To some extent, this definition of a successful team is the notion of classic TF problem and has been adopted by huge body of work in various fields. A classic TF problem can be described as:

> *"The aim to find a team of individuals that satisfies the requirements of a given task while possessing the ability to communicate effectively."*

This problem has been extensively studied in a variety of fields ignoring the social ties between individuals. One area which this occurs is Operation Research (OR). TF problem in this field is formulated as an integer linear program. The aim of these studies is to find an optimum match between people and the demanded functional requirements. The problem is often addressed using techniques such as Branch-and-Cut [146], Simulated Annealing [16] or Genetic algorithms [132]. These studies ignore social ties among individuals and mainly focus on the skills they possess to cover the required skills of a task accordingly. Thus the approaches taken to address TF problem in this field does not properly include the quality of a successful team with respect to an effective communication.

There have been a few studies that aim to address the gap of an effective communication requirement mentioned earlier. These studies take advantage of interdisciplinary concepts such as psychological and interpersonal attributes. One example of this line of work is the person–group fit paradigm proposed in [131]. This paradigm is composed of two elements, supplementary fit and complementary fit. Supplementary fit targets groups with similar personality trends whereas complementary fit emphasises on those people who have different characteristics but can complement with each other to accomplish a task. Although authors suggest that modelling team formation based on their proposed paradigm can lead to a desirable outcome, it is unclear how the personalities of people are identified and implications become practically effective.

Another example of interdisciplinary approaches in classic TF is the study in [33] where authors consider individual drive among the members of a team as an element of a successful team. They employ the *Myers-Briggs* test to measure the personality among candidates and arrange the desired team based on their individual and social characteristics. Following the same motivation, Fitzpatrick *et al.* [54] introduce a parameter based on a technique called *Kolbe Conative Index* (KCI) in order to evaluate the motivation of the team given people's personality and form teams accordingly. The more recent example of interdisciplinary approaches toward TF is the scheme proposed by Zhu *et al.* [145]. Authors propose a role-based TF and claim that their approach will facilitate the complexity of TF in the context of software engineering.

Although the above-mentioned approaches are theoretically intriguing, they fail to address TF in the context of social networks. The importance of underlying social ties between individuals for TF problem became evident by Gaston *et al.* [60]. Authors in this study demonstrate that adapting social network structure in TF significantly improves organisational efficiency. Later in [11] authors study the dynamics of TF and their impact on the formation of communities in social networks. They also find underlying social structure between people an important factor in forming teams. Cheatham *et al.* [32] expand the idea of using social networks in TF to a more practical form by providing two different views and raising realistic questions with respect of forming collaborative teams than can be answered using Social Network Analysis.

The three inspirational studies mentioned above open a new domain for studding TF in social networks, particularly in the field of computer science. They conclude the evolutionary journey of Team Formation in its classic era. This era has been very productive in terms of algorithms devised to study TF in theory and has provided a solid background for researchers today. The next section is dedicated to reviewing the studies of Team Formation in social networks in a modern era.

2.2 Team Formation in Social Networks

The emergence of social networks and wide use of this new technology has provided a new framework for researchers to work on. This has been of particular interest for researchers in the field of computer science who aim to understand the process of social collaboration, that is, how people use social connections to form teams. Modern TF considers collaboration between team members within their social ties a mandatory quality of a successful team. Thanks to the online communities such as DBLP [1], IMDB [2] or online social networks such as Facebook [3], Twitter [4], today researchers can analyse the social structure between people and investigate the new possibilities in forming collaborative teams.

The use of social networks to address TF as a computational problem began with Lappas *et al.* [86]. This influential work was the first attempt to formulate the TF problem given the social structure between individuals. They define TF in social networks as:

> *"Given an underlying social network, the aim of team formation is to find a group of individuals who can communicate effectively as a team to accomplish a specific task."*

They prove that the problem is *NP-hard* and propose a two-step approximation method to form effective teams of experts. The approximation techniques used in this work have been very popular and expanded by many studies since. Despite being a pioneering work, the study in [86] fail to properly address TF problem in social networks with respect to effective communication within the team. The reason is that the definition given for TF in this work heavily relies on the communication cost function. Albeit clearly stating that *"other notions of the effectiveness of a*

[1] www.dblp.uni-trier.de
[2] www.imdb.com/interfaces
[3] www.facebook.com
[4] www.twitter.com

team can lead to a different optimization functions", the authors mainly concentrate on finding teams with minimum communication cost and ignore the other aspect of effectiveness of a team in TF problem. Since no argument has been provided to justify the minimum communication cost as an effective measure to increase the performance of the team, it is debatable whether considering minimum communication cost as effective communication is a valid assumption. This has lead researchers define the TF problem according to their variables.

The divergent vision toward TF is evident from the various parameters and implications used to address the problem. In the next section, I provide an overview of the studies conducted in the area of modern TF with respect to these parameters.

2.3 Types of Team Formation

The success of a team is not associated with a single factor. There is a consensus among researchers that each team behaves differently toward various factors hence the success of a team is relative to their behaviour which also varies according to the circumstances. [84]. Lappas *et al.* [87] survey various approaches of TF in social networks and highlight some of the indicative algorithms in the literature. They outline various aspects of a successful team which have been considered and employed in TF by researchers. In another research, Wang *et al.* [129] and [130] conduct a comparative study on some of these research and categorise them according to the schemes they suggest. Although this comparative study mentions other parameters, the main focus of this work is on communication cost and other aspects of TF are generally dismissed.

For the purpose of this thesis, I review the existing body of research on TF in social networks from the prospective of the parameters used to define the main problem. These parameters have been named differently, thus I categorise them based on the functionality or the purpose

they serve for. These parameters include *Communication Cost*, *Workload*, *Density*, *Expert Rank*, *Personnel Cost*, and *Diversity*. In addition to this parameter-oriented classification, I provide an overview of TF applications in various fields.

2.3.1 Communication Cost: One Metric to Rule Them All

Finding teams with minimal communication cost has been the primary goal of TF in social networks in numerous studies. The consensus in this line of work is that the definition of TF given by Lappas *et. al* [86] potentially covers the necessary aspects of TF in social networks. The minimum communication cost which is the sole measure of effective communication in this work has been the main point of interest. The consequent studies mainly concentrate on optimising the communication cost function and define communication cost such that it produces better results. Wang *et al.* [130] classify these studies in four major classes according to the communication cost function they use. They identify four main categories of communication cost including *Rarest First*, *Steiner*, *Shortest Distance*, and *Leader Distance*.

The first class of these algorithms are based on *Rarest First* [86] which is an extension to *Multichoice Algorithms* [9]. The communication cost in this category is defined as the longest shortest path between any experts in the team. The algorithms *MinDiaSol* [97] and *LBRadius* [7] are good examples of this class. One notable characteristic of algorithms of this nature is the ability to form teams rapidly. This is because they tend to take as many skills as possible from a single expert to cover the required skills. It is disputable whether such an approach results in teams with high performance.

The second classification encompasses Steiner algorithms for which the objective of the communication cost function resembles the objectives of the *Steiner Tree Problem* proven to be *NP-hard* [79]. In this cate-

gory communication cost is defined as the weight cost of the minimum spanning tree for a sub-graph formed by the team. The algorithms *En-Steiner* [86], *MinAggrSol* [97], *LBSteiner* [7], and *Connector-Steiner* [89] are most important examples of this class. Algorithms with this nature have been generally found to produce teams with small sizes. This is because of the greedy nature of these algorithms. One controversial criticism towards these algorithms though is that the density between team members in this algorithm is very low. The reason for this is that these algorithms prune the communication channels and reconnect them using intermediate nodes as terminals thus the number of the channels reduce in overall. The obvious drawback these algorithms suffer from is that they are slow to converge and form teams in a considerably high time-period.

The third category is called Shortest Distance and algorithms of this class define communication cost as the sum of all shortest paths between any two experts in the team. This definition of communication cost has been first suggested in [76] and [78] and adopted by many researchers in the area of TF. The reason for this much influence is that algorithms with this nature are capable of generating teams with relatively small size in a relatively short time. However it is disputable whether such a definition of communication cost is reflective enough of the reality.

Finally, communication function for Leader Distance algorithms is defined as the sum of the shortest paths from all experts to the team leader. The communication function defined in this line of work is an extension to the function defined in Shortest Distance and has been considered in [76], [103] and [124]. There have been a lot of discussion over the use of leader since it's introduction in [77]. Authors in [89] and [90] use this concept in an algorithm called *Enhanced-Steiner* for generalised tasks. Their algorithm is based on the original *Steiner* tree while considering leader. Hashemi *et al.* [103] also expand the concept of leader by proposing discriminative methods to find leaders and form a team accordingly. They suggest that the level of contributions people put towards relationships

in social network are not evenly distributed thus the strength of these collaboration must be considered while forming teams. Authors in [124] argue that a single leader is unable to manage large teams and suggest a parameter named *communication load constraint* to limit the number of team members. They also present a framework to find an optimal team, under the communication load constraint.

One problem with all the above mentioned studies is that their proposed communication cost function is not accurate enough to be applied in real-life examples. This is because the relationships between individuals in reality is multifaceted and asymmetric [45] and [50] thus the communication cost is expected to reflect multiple values dynamically given each facet of the relationship. The idea of such a communication cost function has been vaguely realised in [138]. This work studies the temporal annotations of historical communications in social networks and suggests that the pattern of communication between experts in social networks are different and sensitive to the cost of communication. Although the authors in this work agree that the defining communication cost as a static measure is problematic in practice, they do not provide any practical solution to address the problem.

To cover this gap authors in [30] consider the relationships between individual asymmetric and introduce a parameter to quantify the cost of including an expert into a team. There have also been a few attempt to address this problem using *Semantic Analysis* [66]. Zhou *et al.* [143] use *Topic Models* [121] to label teams based on information sharing activities among the experts. They aim to generate team profiles by which they can reach to the most desirable experts for a given tasks and form teams accordingly. Based on the same technique, authors in [93] present a model to determine the general closeness of the past experiences of experts with the topic and aggregating the rank of experts accordingly. Therefore they estimate the degree to which a group is a knowledgeable for a given topic. However it is unclear what the topics represent as they do not differenti-

ate between required skills and topics. In addition, it is debatable whether their proposed method is practical. This is because the queries for TF problem contains required skills rather than topics. To address this, Zhu *et al.* [144] suggest a scheme to generate categories based on the dependencies between skills and interaction between experts. Based on these categories, they propose a framework for TF to rank the authority of experts and form teams accordingly. Unfortunately none of these studies address the dynamic communication cost requirement explained earlier. This problem has been never approached by any work in the area of TF.

Apart from the necessity of multinomial relationships and dynamic communication cost, the general problem with all the studies in this category is that the requirement of minimum communication cost is so strict that leaves no room for considering other characteristics of a successful team. Having this restriction in mind, some studies propose schemes to relax the skill coverage requirement so that other parameters can be included. One example is the work in [88] where authors propose a method to form teams according to user-specified information. They specify subjects, group size, must-have skills as the customisable body of the query and devise a greedy approximation algorithm to solve TF problem. Another example is given in Bhowmik *et al.* [18] where an unconstrained sub-modular function is proposed. This function relaxes the skill coverage requirements to a "must have" and "should have" level. The authors maximise this function and optimise TF with respect to communication level. The effectiveness of these works is contentious as they aim to alter the problem rather than proposing a practical solution for it.

2.3.2 Workload: The Art of being Fair

A balanced workload between team members is considered an important factor of a successful team [82]. The intuition behind such an idea is that adjusting the amount of work according to the capacity of each mem-

ber of the team provides a fair environment and increases the teamwork. Based on this factor, there have been a few studies which tried to include workload to distribute the skills fairly among the team members though research conducted regarding this parameter in TF is limited and requires more attention. Anagnostopoulos *et al.* [6] and [7] consider the workload of the chosen experts and proposed an algorithm to enable finding teams of experts where the workload is balanced. With the same motivation in mind, Majumder *et al.* [97] take a slightly different approach to the problem by setting a limitation for the workload each member of the team can handle. The authors introduce a parameter called *packing constraint* and control the formation of team such that the workload of the chosen experts do not exceed the packing constraint.

2.3.3 Density: The Closer the Better

Density is a concept derived from *graph theory* which determines the number of edges to a vertex [37]. The motivation behind using density in forming teams is that the teams with higher density are more interconnected hence their communication channels are abundant.

Density was first adopted for TF in social network by [119] based on the minimum degree and distance constraints. The authors first prove that TF with density maximisation function is still *NP-hard*, then they propose a greedy algorithm to calculate density. Finally, they present two heuristic algorithms to find communities of a desired upper density. Gajewar *et al.* [57] continue this line of work by proposing a set of approximation algorithms aiming to maximise density of the team. They equip their algorithms with heuristic extensions in order to create a balance between the size of team and its density. Rangapuram *et al.* [109] expands the use of density by proposing three considerations including leader, cost limitations, and locality of the team. Their approximation technique in the experiment is claimed to produce more coherent and compact teams.

2.3.4 Expert Rank: Finding the Wisest

The essence of ranking experts while forming teams was first realised by Sharifi *et al.* [116] and has since been used in various studies as a metric of a successful team. Farhadi *et al.* [51] propose a framework in which the skill grade of experts are determined. They incorporate these skill grades with minimum communication cost and generate a compound cost metric by which the formation of teams is achieved. The problem with this approach is that the proposed parameter for quantifying expertise does not change according to the skills experts possess whereas in reality it is expected that multi-skilled experts would demonstrate various level of expertise within their skill-set.

To address this problem, Bozzon *et al.* [27] focus on finding individuals with the highest rank for a given task based on the activities of social network users. In order to achieve this, they assert expertise by tracing their activities in social networks and building up profiles for experts. This is done using text analysis and semantic annotation. Awal *et al.* [10] approach the expertise as a metric in TF slightly differently. They argue that forming teams based on the total expertise of a team instead of ranking individuals produces more desirable results. They propose an approach based on *Collective Intelligence* (CI) to recommend candidates who can increase the overall expertise of the team. In addition, they introduce a trust-based collaboration score to enhance the collaboration between experts and facilitate the exchange of ideas and expertise.

2.3.5 Personnel Cost: Thinking of the Budget

There is almost no doubt that there is a strong correlation between the budget of a project and the cost of labour. Hence, as metric of a successful team, personnel cost has drawn attention attention among researchers. Authors in [78] and [5] formulate this parameter and combine it with communication cost to address TF in social networks as a bi-

objective problem. The aim of this work is to find financially affordable teams with minimum communication cost. Stallings *et al.* [120] suggest that the cost of experts are driven from their productivity and impact on the social network. Hence they first discuss the simulation of personnel cost using *h-index* [68] then propose an automatic approach to assign relative credits to experts and argue that their suggested parameter projects less objective cost values in comparison to *h-index*. Golshan *et al.* [64] approaches TF problem from financial perspective too. They propose heuristic approximation to find a financially affordable team of experts in which the total benefit of the projects this team can collectively cover is maximised.

2.3.6 Diversity: Where Everyone Has a Share

Diversity among team members has a positive impact on creativity and contributes to desirable outcomes. [133]. In the context of TF and given the social network of experts where skills are abundant, diverse group of expertise can potentially accomplish the project effectively [70]. Having this in mind, Tong *et al.* [125] adopt diversity as metric and propose an algorithm to find optimum teams with maximum diversity. This work is improved in [137] by truncating the social influence of less desired experts on the diversity of the team throughout the social network. This is accomplished by quantifying diversity based on the social network between potential candidates who can satisfy the required task. Xu *et al.* [136] propose a model to include the skill diversity among the members of a team and formulate diversity in the context of TF. Buccafurri *et al.* [28] propose a method based on collective intelligence to increase diversity of the team during team formation process. Wu *et al.* [134] study TF problem by considering *Diversity of Opinion* as a requirement element of a team in the field of crowd-sourcing. Quantifying this element as metric, they propose two models to find a desirable team of experts for a given task.

Table 2.1: Summary of TF Algorithms Based on the Parameters Employed

Algorithms	Com. Cost	Work-load	Density	Exp. Rank	Pers. Cost	Diver-sity
[30], [76], [86], [88], [89], [90], [93], [124], [138]	✓					
[6], [7], [97]	✓	✓				
[57], [119]			✓			
[27], [119]				✓		
[51], [103], [144]	✓			✓		
[64]					✓	
[5], [18], [78]	✓				✓	
[109]			✓		✓	
[10]	✓			✓		✓
[125], [134], [137]						✓

To summarise this section, Table 2.1 outlines the factors employed for Team Formation throughout the literature. In the next section, I discuss the applications of team formation in various research areas.

2.4 Team Formation Applications

Today, the concept of team and teamwork is an indisputable part of every project in every field. Hence TF has been considered in various areas of research as an application to facilitate the process of finding potentials according to the criteria. The vast number of these field of studies makes

it extremely hard to categorise various schemes according to applications they have been adopted. For example in the field of *education*, Agrawal *et al.* [2] use TF to maximise the gain of students by team partition. In the field of *organisational management* authors in [91] set up events as tasks and people as experts and adapt the existing greedy heuristic algorithms proposed for TF problem in order to find attendants for a given event. Another example is the study in [3] where TF problem has been accommodated for the field of manufacturing and a parallel hybrid grouping genetic algorithm to solve the problem of *Machine-Part Cell Formation* has been propose.

Albeit being subject of study in almost every field, TF has been extensively applied in three fields in particular. These fields include *Crowdsourcing*, *Recommendation Systems*, and *Multi-Agent Systems*. The rest of this chapter has been devoted to exploring the use of TF in these areas.

2.4.1 Crowdsourcing

The term "crowsourcing" was first introduce by [71] and defined as "*the act of taking a job traditionally performed by a designated agent (usually an employee) and outsourcing it to an undefined, generally large group of people in the form of an open call*". Since then various crowdsourcing applications such as *Wikipedia* [5] has been developed and become extremely popular. Besides, the use of TF in crowdsourcing has drawn a lot of attentions between scholars.

Cao *et al.* [31] use the concept of TF to select appropriate juries for decision making task on micro-blog services. They refer to this problem as *Jury Selection Problem* (JSP) and address it by adjusting the existing TF algorithms according their requirements. This work is later expanded by Zheng *et al.* [142] where the *Bayesian Voting* (BV) is employed to find juries with the highest rank. They consider answers for a given task as

[5] www.wikipedia.org

posteriors and task owner's beliefs on the answer as priors. Another study in the area of crowd-sourcing is conduced in Zhao *et al.* [140] where the process of crowd selection on micro-blogs has been automated thanks to the query processing techniques. Authors expand this research in [141] where they use probability models to transfer the given task into a defined set of skills and perform crowd selection accordingly. Finally in [139], the authors approach the TF problem in the context of crowd-sourcing where the knowledge about past experiences within the experts is limited. They categorise experts based on their past experiences and propose a method in which the expertise of well-known experts is inferred and distributed among every expert in the social network.

2.4.2 Recommendation Systems

Recommendation system are particular class of *Information Filtering Systems* that aim to predict the rating of items or preference of users and recommend items accordingly [111]. These systems can be incorporated with the concept of TF in social networks to facilitate the process of recommendation.

Authors in [43] and [44] argue that measuring direct interaction to determine communication between experts produce inaccurate results for TF. They address this gap by using implicit recommendations of collaboration to support even sparsely connected networks. They provide two heuristics based on *Genetic Algorithms* and *Simulated Annealing* to create a trade-off between skill coverage and team connectivity in team formation. Xie *et al.* [135] utilises the *top-k* TF concept to generate a package of recommendations consisting of multiple items. Each item is viewed as an expert with a cost and the package is considered as a team to be formed. The aim is to assemble items into a package and recommend it to a user according to their requirements and specified budget. Parameswaran *et al.* [107] employ the concept of TF to study the prob-

lem of recommending courses to a group of students based on a set of constraints. In this case, the courses are considered as tasks whereas the target students are considered as the selected team.

The most recent of example of using TF in recommendation systems is suggested in [92] where finding a substitution for a team leader in social network has been studied. Authors in this work define a problem called *Team Member Replacement* and propose an algorithm to replace an expert who is leaving the team with another expert based on the interaction between skills and matching structure. Motivating by the same idea, Fitsilis *et al.* [53] propose a system in which the collective knowledge obtained from social networks is analysed to potential candidates for a given task are recommended.

2.4.3 Multi-Agent Systems

The use of TF in Multi-Agent System (MAS) was first suggested by Gaston *et al.* [59]. Authors in this study demonstrate that finding appropriate social network structures between agents has a significant impact on the performance of the overall system. Since this finding, the concept of TF in social networks has been extensively adopted in MAS.

Bulka *et al.* [29] adjust the TF concept for socially connected MAS and describe a policy learning framework for forming teams based on local information. Bai *et al.* [12] provide a guideline of using TF in self-interested multi-agent systems particularly in dynamic environments where task requirements and resources change regularly. They argue that in such an environment, a selfish agent might modify or replace the collaboration relationships with its teammates. To avoid this, the authors propose a flexible TF mechanism that can enable agents to choose team members with reasonable terms. Corgnet *et al.* [39] adopt the concept of positive learning and develop a model of TF for multi-agent systems in which agents are self-aware. This model generates divergence in team forma-

tion and prevents the agents from asking for excessive shares of the group outcome. Maghami *et al.* [96] introduce an agent-based simulation in order to investigate the impact of social factors such as stereotypes on the formation of task-oriented groups. They derive stereotypes from the agents' past experiences and apply them during the TF process in order to improve the diversity of skills among agents.

2.5 Findings and Summary

In this chapter, I reviewed existing classic and modern literature in the area of Team Formation. I provided a comprehensive survey on these studies and classified the them based on the utilised parameters. I critically approached the most notable studies in each category. Furthermore, I discussed the various applications of team formation in other fields. During this review, I identified four main gaps including the lack of a realistic approach towards TF with respect of the well-defined measures of the success of a team, the multifaceted relationship, the dynamic nature of the communication cost, and fine-grained expertise level of the experts in a given skill.

The first problem with all the studies in the area of TF in social networks is their mono-objective or bi-objective nature. As observed from Table 2.1, the majority of research in the field of TF consider only one or two parameters and ignore other important characteristics of a successful team. Although it is understandable that implementing a vast number of these characteristics is very difficult, an approach equipped with a compound measure consisting of key social and psychological elements of a successful team is achievable. Such an approach can take the focus away from minimising communication cost as an ultimate goal and potentially normalising the TF formation across other parameters.

The second gap is the unrealistic view toward relationships between experts by which the communication cost is driven. It became evident

that all the studies in the field of TF consider relationships nothing more than a connection or past experience. Given that the relationships are multifaceted in real life, it is debatable whether such an assumption is practical and reflective of real-life scenarios. Although some efforts has been made to consider hidden aspects of TF in social networks using *semantic analysis* tools, the main focus has been to expand the personal attributes of experts rather than interpersonal relationships between them.

The third gap I identified is the definition of communication cost as an independent and static parameter. Apart from the problems caused by minimising communication cost, assuming relationships as monofaceted connections leads to a belief that the communication cost between experts is a static component of TF and independent from the task. In reality though the scale of communication highly depends on the common grounds between people which dynamically changes according to the circumstances. In other words, the ability to communicate with one another highly depends on the knowledge and expertise of both ends of communication. Hence, if the subject changes, the ability of the communication must change accordingly. This is an important attribute of communication which has not been realised nor addressed in any of the reviewed studies.

The final gap I discovered in this literature review is the insufficient and inaccurate parameters defined for measuring expertise. While a few studies admit expert rank as an indispensable factor in the success of a team, the approaches taken by the researchers in the area of TF in social networks toward employing this factor are questionable. These approaches mainly have a general and vague view of expertise discarding the skill-set of the experts. Given that people possess different level of expertise within their skill-set, it is questionable whether such a general view is practical.

In the following chapter, I aim to address the gaps mentioned above by proposing a novel approach called Chemistry-Oriented Team Forma-

tion (ChemoTF). I first address the multinomial nature of relationships by a novel definition of a relationship and elucidate its building blocks. In addition, I redefine TF in the presence of some novel parameters including *Dynamic Communication Cost, Chemistry Level, Expertise Level, Expert Cost.* Having formally defined all the concepts, I present an analytical model and an algorithm for my approach.

"Computer Science is a science of abstraction - creating the right model for a problem and devising the appropriate mechanizable techniques to solve it."

Alfred Aho [4]

Chapter 3

Chemistry Oriented Team Formation

The lack of an accurate and effective communication model in the definition of team formation leads to four main gaps including: the lack of a realistic approach towards TF with respect to the well-defined measures of the success of a team; the multifaceted relationship; the dynamic nature of the communication cost; and fine-grained expertise level of the experts in a given skill. Addressing these gaps requires a novel approach and a well-designed mechanism in order to form cost-effective and communicative teams with highest level of expertise possible.

This chapter elaborates an approach called Chemistry Oriented Team Formation (ChemoTF) to addresses team formation problem in social network based on various novel concepts and parameters. First, these concepts and parameters are formulated and TF problem is formally declared accordingly. Then based on these definition, an analytical model for ChemoTF is illustrated and its mechanism of how the TF problem is tackled is thoroughly explained. Finally a formal algorithm for the proposed approach is presented and the process is elaborated. To the best of my knowledge, this is the first attempt to address team formation problem in social network using a chemistry-oriented approach.

3.1 Team Formation Problem

The aim of TF is to find experts who are qualified to fulfil the task and can communicate efficiently in a cost effective manner. Given that there is no consensus among researchers about the quality of teams with respect to ideal level of expertise, communication, and cost, TF has led to various interpretations. In order to give a clear understanding of the problem, I formally declare the principal TF elements and introduce parameters to reflect team quality. Then I provide a formal definition of TF problem accordingly. The elements include Multidimensional Social Network and its components. The parameters include *Chemistry Level, Dynamic Communication Cost, Expertise Level, Expert Cost.*

3.1.1 Key Definitions

In this section, I define the principal TF elements and parameters.

Definition 3.1.1. (*Multidimensional Social Network*)

Let $X = \{x_1, x_2, \ldots, x_m\}$ be a pool of experts, each possessing a set of skills $x_i \subseteq S$. I define Multidimensional Social Network $H = (X, R, Z)$ as an undirected multidimensional weighted graph. It is an extended form of $G = (V, E)$ (V represents a universe of experts and E represents a universe of experiences between experts) which is filtered according to T using a $SetCover$ function such that $X \subseteq V$. Each vertex x_i represents an expert who at least possesses one skill s_a such that $s_a \in T$ if $SetCover(s_a)$ is applied. Each edge $r_{i,j}$ represents a relationship between a pair of experts which is a multinomial probability distribution of experiences between x_i and x_j. The weight assigned to each edge represents dynamic Communication Cost between the pairs which varies according to the dimensions in the universe of K dimensions $Z = \{z_1, z_2, \ldots, z_K\}$. I call each z in this universe an Area of Expertise and define it as a probability distribution of n skills over relationships throughout H such that $z_i = \{P_{s_1}, P_{s_2}, \ldots, P_{s_n}\}$ and $\sum_1^n P_{s_n} = 1$.

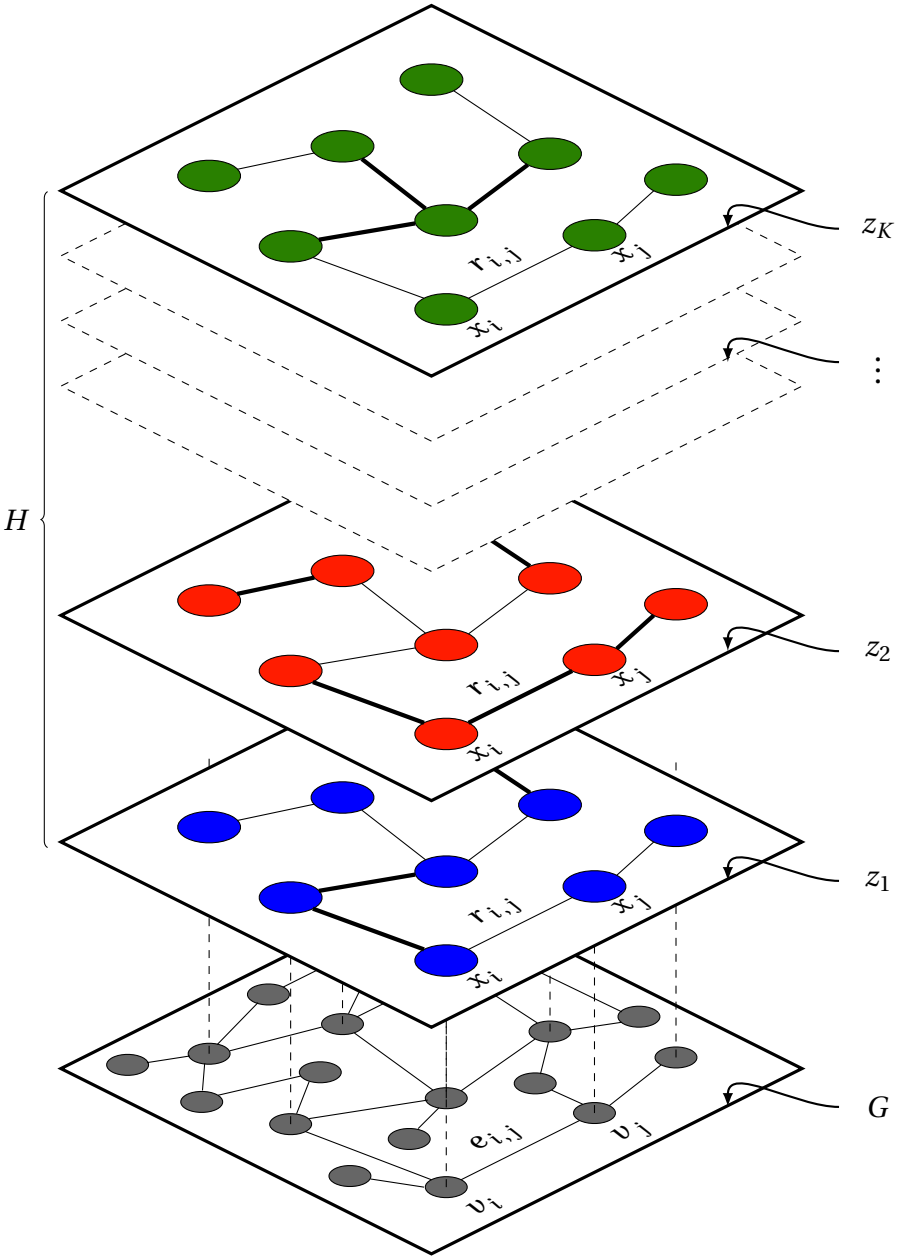


Figure 3.1: An Example of a Multidimensional Social Network

Definition 3.1.2. ***(Task)***

A task $T \subseteq S$ is defined as a subset of skills required to perform a job.

Multidimensional Social Network provides multiple views over the relationships thanks to the multinomial nature of relationships and areas of expertise considered as the dimensions of this network. As observed from Figure 3.1 each area of expertise represents a different outlook and dispense a unique perspective among relationships throughout the network. This suggests that the relationships in such network are multinomial and there are K relationships $r_{i,j}$ as well as K Communication Cost value between x_i and x_j. This allows the connections between expert to remain in tact whereas the Communication Cost changes depending on the selected area of expertise z. In other words selected area of expertise z determines the value of Communication Cost between experts which can be high (marked by thick edges) or low (marked by thin edges). This is the principal of Dynamic Communication Cost which is defined as follows:

Definition 3.1.3. (*Dynamic Communication Cost*)

I define dynamic Communication Cost ($DComCom$) between x_i and x_j as the extent the pair share the same areas of expertise z within their relationships and calculate it as follows:

$$DComCost(x_i, x_j) = \sum_{k=1}^{K} P(x_i|z_k)P(z|r_{i,j}) \sum_{k=1}^{K} P(r_{i,j}|z)P(z|x_j) \tag{3.1}$$

Note that the Definition 3.1.3 assumes that each expert possesses a single skill. There are two reasons for such an assumption, both lie on the nature of the ChemoTF algorithm. The first reason is that this algorithm works in rounds, each observe experts with a single skill. In order to calculate DComCost for all the skills of the pairs, ChemoTF has been designed to work in loops. The second reason for such an assumption is that, in each round of ChemoTF, I compare Dynamic Communication Cost of the pair with Chemistry Level of the skills used by the pair. Since Chemistry Level accepts only two skills as input, Dynamic Communication Cost should be compatible for such a comparison. Definition 3.1.4 delineates the concept of Chemistry Level between a pair of skills.

Definition 3.1.4. (*Chemistry Level*)

I define Chemistry Level ($ChemLvl$) between s_a and s_b based on the definition of document similarity [121] as the extent the pair share the same areas of expertise z within all the relationships throughout the network given their skills and calculate it as follows:

$$ChemLvl(s_a, s_b) = \sum_{k=1}^{K} P(s_a|z)P(z|s_b) \tag{3.2}$$

Comparing the Equation 3.1 to Equation 3.1, it is observed that the skill domain in Dynamic Communication Cost is the universe of all skills S, whereas this domain in Chemistry Level has been defined as the Task T. The reason for such definition is that the inputs for DComCost() function are experts with multiple skills. This means that the experts in this case might have skills such that $\{x_i, x_j\} \not\subset T$. However the inputs of Chemistry Level are skills such that $\{s_a, s_b\} \subseteq T$. Furthermore, it is evident that Dynamic Communication Cost is an extended form of Chemistry Level for a specific relationship $r_{i,j}$. In other words, given a pair of required skills, Dynamic Communication Cost reflects the communication cost involved between a pair of expert to fulfil the task whereas Chemistry Level mirrors the expected level of communication required to perform it. This comparison is one of the criteria of an effective team in my definition of TF which insures the selected experts can communicate in an expected necessary level. Other criteria include Expert Cost and Expertise Level which I define as follows.

Definition 3.1.5. (*Expert Cost*)

I define Expertise Cost ($ExpCost$) of an expert $x_i \in V$ based on the definition of $h-index$ [68] as the sum of h experiences $e_{i,j}$ or $e_{j,i}$ in which x_i has been part, on the condition that the experience has at least h impact throughout the network:

$$ExpCost(x_i) = \bigvee_{i=1}^{h} x_i \wedge i \tag{3.3}$$

Definition 3.1.6. (*Expertise Level*)

I define Expertise Level ($ExpLvl$) of an expert $x_i \in V$ in skill s_a as the sum of all experiences $e_{i,j}$ or $e_{j,i}$ in which x_i has been part of, on the condition that the experience has been associated with the skill s_a. Given the degree of x_i denote by $d_i = deg(x_i)$, $ExpLvl(x_i|s_a)$ is calculated as bellow:

$$ExpLvl(x_i|s_a) = \sum_{n=1}^{d_i} [s_a \in x_i] \tag{3.4}$$

Definitions 3.1.1- 3.1.6 encompass all the TF elements and concepts. A summary of the notations used to define the concepts and the interpretation of each notation have been demonstrated in Table 3.1. Having defined and clarified all the TF element, I formally state the Team Formation Problem in next section.

3.1.2 Problem Statement

Given Definition 3.1.1- 3.1.5 stated in the previous section, I formally define Team Formation Problem as follows:

Definition 3.1.7. (*Team Formation Problem*)

Given a a Multidimensional Social Network $H = (X, R, Z)$, a set of experts X each with a finite number of skills $s \in S$, and a Task $T \subseteq S$, find $X' \subseteq X$ such that:

$$\begin{cases} (1) & X' \cap T = T \quad and \quad |X'| \leq |T| \\ (2) & sumDComCost(X') \leq sumChemLvl(T) \\ (3) & sumExpCost(X') \leq Cost(T) \\ (4) & sumExpLvl(X') = Max \end{cases}$$

As observed from the Definition 3.1.7, TF problem is described as the problem of finding a cost-efficient team with highest expertise that can satisfy the required skills of a task and can communicate in an effective

Table 3.1: Notation of TF elements

Notation	Interpretation
S	A set of specified skills
T	A task containing required skills
V	A set of experts in social network
E	A set of experiences between a pair of experts
$G(V,E)$	A social network of experts
X	A set of experts in H with required skills
R	A set of relationships between a pair of experts
Z	A set of areas of expertise
$H(X,R,Z)$	A multidimensional social network of experts
$DComCost(x_i,x_j)$	Dynamic Communication Cost of a pair of experts
$ChemLvl(s_a,s_b)$	Chemistry Level of a pair of skills
$ExpCost(x_i)$	Cost of an expert if hired
$ExpLvl(x_i\|s_a)$	Expertise Level of an expert in a specific skill

manner. The first rule in this definition determines that the selected experts must cover the required skills of a given task. In addition it imposes that the number of selected experts in the team (cardinality of the team) must not exceed the number of the skills of a given task (cardinality of the task). The second rule stipulates that the cost of communication between experts in the selected team must meet the expected level of communication required to perform the task given its skills. The third rule guarantees that the cost of the selected team is always lower than the expected cost of the task. The final rule ensures that the selected team has the maximum level of expertise.

TF problem has been proven to be *NP-hard* [86]. Therefore there have been numerous efforts to address this problem using various techniques. However as mentioned in the previous chapter, almost all of these studies aimed to minimised the communication cost of the team ignoring the

other aspects of an effective team in TF problem as as formulated in Definition 3.1.7. To the best of my knowledge, this thesis is the first attempt to achieve this goal using a novel approach called Chemistry-Oriented Team Formation. The next section is dedicated to elaborating this approach.

3.2 Chemistry-Oriented Team Formation

In order to tackle TF problem, I introduce an approach called Chemistry-Oriented Team Formation (ChemoTF). This approach is fundamentally different than previous studies in the area of TF. The most significant difference is that it is based on the positive chemistry between experts rather than the distance between them. Such an approach will take away the emphasis from finding teams with minimum communication cost and makes it possible to consider other aspects of the quality of a team. In other words, focusing on finding teams with minimum communication cost can in turn lead to the neglect of those experts who have higher levels of expertise but have not been included in the search criteria due to restrictions applied by Communication Cost function. This can result in a lower overall expertise level. Given that the nature of some tasks require people with as much as expertise as possible, ChemoTF provides a unique yet efficient measure called Chemistry Level formally introduced in Definition 3.1.4 to include teams with necessary communication cost and create a balance between Dynamic Communication Cost and Expertise Level.

The second important and unique feature of ChemoTF is its realistic and accurate approach toward TF problem. This is achieved by defining Communication Cost as a dynamic variable which changes according to various dimensions of the multinomial relationship between experts as formally declared in Definition 3.1.3. The nature of relationships between people demands the scale of their communication to be more than just a number but a variable across various facets of the their relationships.

Thanks to areas of expertise serving as the dimensions of multidimensional social network and the multinomial relationships between experts, ChemoTF derives Dynamic Communication Cost of experts from their relationships according to the skills they have used during their mutual experiences.

The third distinctive attribute of ChemoTF is the ability of forming teams with maximum level of expertise possible. This is accomplished by the novel parameter called Expertise Level defined earlier in Definition 3.1.6. The notion of this parameters lies on the assumption that the more expertise within the team leads to better overall level of productivity. Using Expertise Level, ChemoTF can find better teams with respect of expertise, the quality which is surprisingly neglected in previous studies.

Finally, ChemoTF has the ability to identify the costly teams and filter them according to the expected cost of an expert given his or her skills. This is achieved using a personnel cost metric called Expert Cost defined earlier in Definition 3.1.5. Given that teams with high Expertise Level can be very costly, ChemoTF takes Expert Cost to insure the selected experts are not unusually expensive. Comparing this value with an average cost of a skill throughout the social network returns teams with high expertise yet with a reasonable personnel cost.

To this end, ChemoTF aims to tackle TF problem by answering the following question:

How to address TF problem in a more realistic fashion to form teams with maximise expertise level of experts who can satisfy the requirement of an effective communication within a desirable cost?

The rest of this chapter has been dedicated to elucidating the architecture and mechanism of ChemoTF. First an analytical model has been illustrated and its architecture has been discussed. Then an algorithm for ChemoTF is given and further explained.

3.2.1 Analytical Model

I introduce ChemoTF by providing an analytical model to outline its structure. Figure 3.2 illustrates this model in a step-by-step process in which expert are processed through a set of steps and the most desirable experts to fulfil the given task are returned as an expert team. As demonstrated in this figure, the assumption is that there is a social network consisting of all the experts, a task which needs to be performed, and a multidimensional social network drawn from social network using a $SetCover()$ function such that each expert holds at least one skill from the task. These three elements which are illustrated on the left side of the given figure constitute the inputs of ChemoTF.

The scheme begins by choosing an expert randomly from the multidimensional social network and adding the expert to the team. The reason behind adding expert to the final team in the first step is that $SetCover()$ function is applied during the generation of multidimensional social network. This function filters the multidimensional social network according to the skills in the task and guarantees that the chosen experts have at least one required skill and all the consequent actions are valid. This step is taken for each skill in the task.

The scheme continues in step 2 and 3 by drawing a sub-graph of all those experts that are socially connected to the previously chosen experts and choosing an adjacent expert randomly. Then in step 4, the pair of already added expert and their adjacent expert are taken to a four sets of comparisons in step 5. These comparisons are the notion of ChemoTF and assess the quality of team based on the four rules of TF defined in Definition 3.1.7. The first comparison in step 5.1 ensures that the selected adjacent expert can cover at least one skill that the already added expert cannot. This comparison guarantees fulfilment of the first rule of Team Formation. The second comparison is performed in the step 5.2 where the Dynamic Communication Cost and Chemistry Level are calculated and compared. This comparison ensures that the Communication Cost

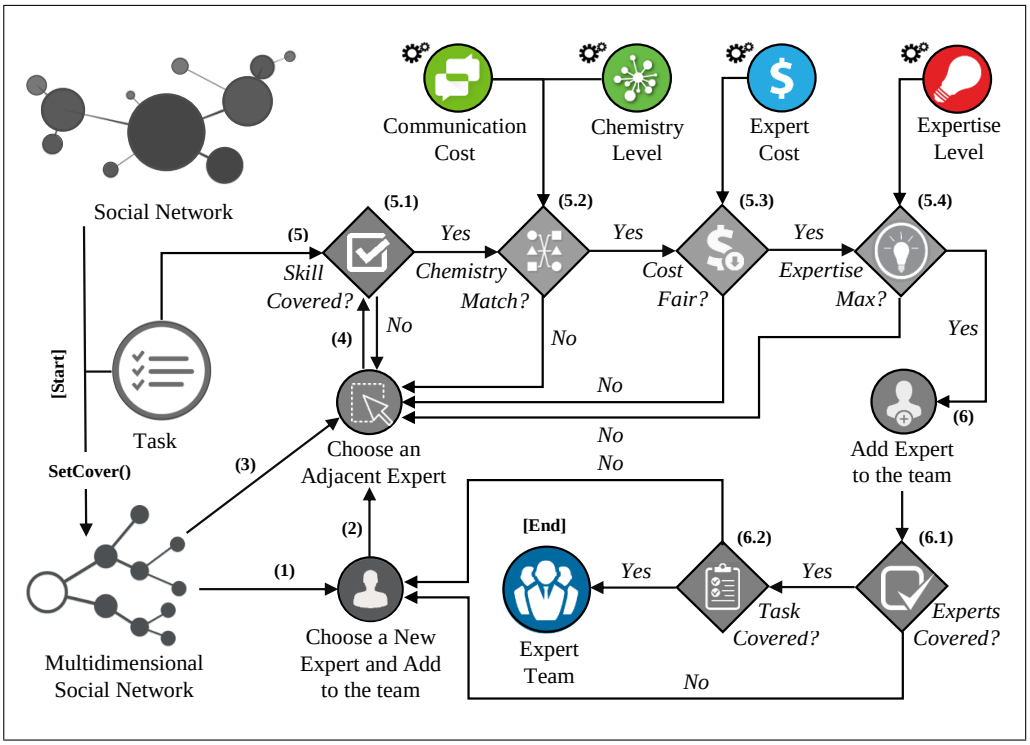


Figure 3.2: ChemoTF step-by-step Analytical Model

of the pair fulfils the expectations of an effective communication. The pair who cannot satisfy the criteria are ignored and another adjacent expert is chosen. This process is repeated until a desirable pair is found.

Step 5.3 promises the fulfilment of the third rule of TF which is to find a cost-effective team. First, Expert Cost of the pair is calculated. Then the sum of this cost is compared to the sum of the average cost of the experts who possess the specified skills throughout the social network. If the sum of the Expert Cost is lower than the projected cost of skills, the pair is taken to the next step, otherwise another adjacent expert is chosen and the control jumps over step 3. Finally the comparison in step 5.4 mirrors the last requirement of TF which is to find a team with maximum Expertise Level. In order to achieve this, sum of the Expertise Level of the pair is calculated and compared to last best team with respect of

the maximum Expertise Level. If the selected pair has a higher Expertise Level than the one in the stack, the best team is overwritten and replaced by the selected pair. This process continues until all the adjacent experts are processed and the team with the highest expertise is found.

Having met all the required rules, the selected pair is added to the final team in step 6. This process continues until all experts and all the required skills are covered (step 6.1 and 6.2). The final *Expert Team* is the output of the ChemoTF. In the next section, I formally present and algorithm for this model.

3.2.2 Algorithm

In this section I provide an algorithm for ChemoTF. The pseudo code of this algorithm is presented in Algorithm 1. The numbering and markings on the left side of this algorithm corresponds to the steps taken in the analytical model of ChemoTF illustrated and explained in the previous section. As observed from the algorithm, ChemoTF takes two inputs namely a pool of experts X and a task T. The assumption is that multidimensional social network H derived from social network G is available. The outputs of this algorithm is a team X' with maximum sum of Expertise Level and a desirable cost (including Communication Cost and Expert Cost). The algorithm starts with drawing a multidimensional social network H from social network G such that the experts have at least one required skill s defined in the task T. This is achieved using $SetCover(T)$. The outcome of this function along with the outcome of $Draw()$ function used later in step 1 and step 4 are obtained from a pre-built hash table in order to improve the efficiency of the algorithm. Then for each required skill s_a the following steps are performed:

At first the pool of experts $X(s_a)$ is populated by the experts who posses skill s_a using the $Draw(s_a)$ function. Then the sum of Communication Cost is initiated to zero. For each expert in the drawn pool $X(s_a)$, Exper-

Algorithm 1: Chemistry-Oriented Team Formation

Input: $G(V,E)$, $H(X,R,Z)$, A pool of expert X each with skills $s \in S$, and a Task T

Output: An expert team X' with maximum $sumExpLvl$ and desirable $sumComCost$ and $sumExpCost$

[Start] $H \sim SetCover(T)$

foreach *skill* $s_a \in T$ **do**

(1) $X(s_a) \leftarrow H.Draw(s_a)$

$sumComCost \leftarrow 0$

foreach *expert* $x_i \in X(s_a)$ **do**

$sumExpLvl \leftarrow ExpLvl(x_i|s_a)$

$sumExpCost \leftarrow ExpCost(x_i)$

(2) $X'.Add(\langle s_a, x_i \rangle)$

(3) **foreach** *skill* $s_b \in T \setminus \{s_a\}$ **do**

(4) $X(s_b) \leftarrow H.Draw(s_b)$

(5.1) **foreach** *expert* $x_j \in x_i.Neighbours(X(s_b))$ **do**

$maxExpLvl \leftarrow 0$

(5.2) **if** $DComCost(x_i, x_j) \leq ChemLvl(s_a, s_b)$ **then**

(5.3) **if** $ExpCost(x_j) \leq s_b.Cost$ **then**

(5.4) **if** $ExplLvl(x_j|s_b) > maxExpLvl$ **then**

$selectedExpert \leftarrow x_j$

$maxExpLvl \leftarrow ExpLvl(x_j|s_b)$

$sumComCost \leftarrow sumComCost + DComCost(x_i, x_j)$

$sumExpLvl \leftarrow sumExpLvl + ExplLvl(x_j|s_b)$

$sumExpCost \leftarrow sumExpCost + ExpCost(x_j)$

(6) $X'.Add(\langle s_b, selectedExpert \rangle)$

[End] **return** $X', sumComCost, sumExpLvl, sumExpCost$

tise Level of x_i with respect to s_a is calculated and added to the sum of Expertise Level of the team. Expert Cost of x_i is also calculated and added to sum of Expert Cost of team. In addition, the selected expert x_i coupled with skill s_a is added to the final team. This action corresponds to Step 2 of previously illustrated analytical model. Having s_a and x_i in hand, step 3 initiates a loop for each skill s_b in the task which excluding s_a. This exclusion insures the search domain do not contain the skills which have already been taken. For each skill s_b in this loop, in step 4 a new pool of experts pool $X(s_b)$ is drawn. In step 5.1, for each expert x_j which is adjacent expert of x_j the following steps are taken to meet TF requirements.

The $Neighbours(X(s_b))$ function in step 5.1 along with the exclusion of the previously taken skill s_a in step 3 satisfy the first rule of TF. The second rule is met in step 5.2 when the Dynamic Communication Cost of the experts x_i and x_j along with the Chemistry Level of s_a and s_b are calculated using Equation 3.1 and Equation 3.2 and compared. If the the cost is satisfactory, the next third rule of TF is put in place in step 5.3. This steps calculates Expert Cost of the adjacent expert x_j using Equiation 3.3 and compares it to the average cost of the skill s_b for which he or she is chosen for. If this cost is also satisfactory, then the last rule of TF is observed in step 5.4. In this step, it is guaranteed that the Expertise Level of the final team is in its highest possible level. In order to accomplish this, first Expertise Level of the adjacent expert x_i with respect to the skill s_b is calculated using Equation 3.4 and is compared to Expertise Level of the team previously kept as the most expert team. If the Expertise level of the new team exceed the Expertise Level of the previous team, then value in the stack is replace by x_j and the maximum Expertise Level is overwritten by the Expertise Level of x_j with respect to s_b.

After all the adjacent experts of x_i are processed and analysed, the sum of Communication Cost, sum of Expertise Level, and sum of Expert Cost are updated and the selected expert x_j coupled with skill s_b is added to the team. This process is repeated until all experts in the drawn pool of ex-

perts $X(s_a)$ and all the skills s_a in the skill are processed. The algorithm returns the best team with respect of Communication Cost, Expertise Level and Expertise.

The time complexity of ChemoTF depends on three factors. The first factor is the number of required skills in the task or simply the cardinality of the task $|T|$. The second factor is the complexity of DComCost function. Since DComCost considers the relationships between experts in various dimension, it has to be calculated for each dimension. Thus the number of Areas of Expertise $|Z|$ is the complexity of this function. The final factor is the number of generated sub-graphs $X(s)$ according to required skill s. This sub-graph is utilised by $Draw(s)$ and $Neighbours(X(s_b))$ in ChemoTF. Creating such sub-graph during the exaction of algorithm can be costly and employing B-Tree [15] indexing can reduce the time complexity to $O(logn)$. In order to achieve an even more efficient ratio, a hash table has been used which is accessible prior to the execution of algorithm. Having resided the hash table in the memory, the time complexity of calculation of parameters becomes $O(1)$ in each call. Using this technique, the worst time complexity of ChemoTF becomes $O(|K|^2 \times |T|^2 \times |X_{max}|^2)$ where $|X_{max}|$ is the maximum number of generated sub-graph of experts possessing a particular skill s.

There is an obvious resemblance between ChemoTF and MinSD [76]. Both algorithms create sub-graphs according to the required skills and explore the neighbouring nodes in a similar fashion. This is the origin of their similar time complexity. From this perspective, ChemoTF can be considered an extension to MinSD which improves the quality of final team with respect of the novel parameters it uses and improves the accuracy of the observations by introducing dimensions to the social network. The difference, however, is the goals set by each algorithm. On one hand, ChemoTF has a positive approach toward TF and aims to find cost effective teams with maximum capabilities and acceptable Dynamic Communication Cost within the range controlled by Chemistry Level. This

is made possible by leveraging connections to multinomial relationships and creating a multidimensional social network accordingly. On the other hand, the goal of MinSD is find teams with minimum Communication Cost regardless of the other attributes of a successful team.

To this end, I addressed Team Formation problem in social networks using Chemistry-Oriented Team Formation approach. To summarise this chapter, I formally defined the elements of Team Formation and stated the TF problem using these definitions in the first section. Then in the second section, I introduced ChemoTF as my solution for TF problem and described it through an analytical model and a formal algorithm. The next chapter describes the methods employed to design and collect the dataset in order to experiment ChemoTF.

> "Big data is not about trying to teach a computer to think like humans. Instead, it's about applying math to huge quantities of data in order to infer probabilities."
>
> *Viktor Mayer-Schönberger* [99]

Chapter 4

Corpus Design and Construction

In order to conduct an experiment on ChemoTF and evaluate the results, a dataset consisting of experts and their social ties is required. Such a dataset requires careful consideration with respect to design so that it becomes effective, practical and reusable. For the purpose of this thesis, I collected extensive amount of bibliographic data from the publications in the field of computer science and compiled a new corpus called Comprehensive Scholarly Corpus (CompScholarCorp). This chapter elaborates the steps taken to collect the data from different sources and studies the design and construction of CompScholarCorp.

The first section provides a general background about various methods with respect to corpus design and construction. The second section elaborates the motivation behind compiling a new corpus rather than employing the existing corpora. In addition, it clarifies the objectives of compiling the new corpus. The third section describes the data sources where the necessary data was collected from. The fourth section delineates the process of sampling and data collection using the mechanism designed and implemented in order to automate data collection process. The final section unfolds the decisions made during the corpus design, presents the designed schema for CompScholarCorp, and illustrates sample records of this corpus.

4.1 Background

A corpus is defined as a systematic and structured collection of naturally occurring texts in electronic format and selected according to external criteria to represent, as far as possible, a language or language variety as a source of data [118]. Corpus is widely used in the field of linguistics but it is also used in other fields. Today, the World Wide Web provides a mine of language data of unprecedented richness and with great ease of access [81]. The technological advancement and availability of large computerised forms of text have transformed the corpus design methodologies and data collection techniques. Fields such as Computational Linguistics (CL) and Natural Language Process (NLP) have emerged and have drawn a lot of attention among researchers who are interested in using corpora as testbeds [73]. This has brought the use of corpus methods to almost every research area including computer science.

Constructing a corpus requires careful considerations with respect to design. There is a large body of research in the area of corpus design and the consensus is that a good design is the one that shows the corpus is valid and serves to the objectives of the research. Having said that, there is no unified model or convention to specify corpus design phases. Authors in [117] provide a general overview of corpora design processes and survey possible phases involved during the corpus construction process. The main aim of all these phases is to make sure the corpus is both valid and representative. The most followed four steps in the corpus construction process include *describing motivations and objectives*, *identifying the source of data*, *specifying sampling criteria and collecting data accordingly*, and *designing corpus structure to present the collected data.*

The first phase of compiling a corpus is to describe motivations and research goals. The notion of this step is to elucidate what motivates the researcher to compile a new corpus rather than using existing corpora. It also involves understanding different types of corpora, and determining

the corpus typology [80]. The second phase usually explicates the source of data, provides an overview of the new corpus, and argues why the new corpus is an authoritative object of study. For a corpus to be authoritative, it needs to be a sample of a language. This takes us to the third phase of corpus construction which is usually referred to as *sampling* and *data collection*. Sampling is a process that includes defining explicit linguistic criteria by which pieces of language are selected and ordered [118]. The criteria themselves are series of hypothesis and restrictions which explain why certain types, number of words, or texts of language need to be collected. This is an important step in constructing a corpus because it determines the data collection strategy. In the case of collecting information from the web, valid sources (seeds) of URLs are identified, possible tool chains are studied, and suitable techniques are adopted to produce an implementation plan for data collection [14]. The last phase is to design a corpus structure in comparable and reusable manner. In this step a decision is made on how the corpus should be formatted and data should be stored [110].

I have dedicated the rest of this chapter to describing how I designed and constructed CompScholarCorp according to the above mentioned four main steps.

4.2 Motivation and Objectives

The process of gathering required information in a systematic fashion can be both time-consuming and challenging. It is recommended to use existing corpus or a combination of multiple corpora [118]. In the case of this research, the existing corpora in the literature do not mirror the required information. An ideal corpus for my thesis must contain large number of experts with their designated skill-sets and the extent of collaboration between experts. These are vital pieces of information from which the expert network is constructed, the relationships are composed,

the metrics are calculated, and finally the results of the ChemoTF are generated. To the best of my knowledge, no work has ever compiled a corpus in this scale with the above-mentioned characteristics.

Various dataset such as LinkedIn [1] , DBLP [2] or IMDB [3] have been employed to tackle TF but probably the most popular dataset throughout the literature has been DBLP. It is a bibliography dataset that provides information about publications in the field of computer science. A co-authorship network where two authors are connected if they publish a specified number of papers together has been constructed and used in various scholarly work [51,76,78,90,97,124,129,130,137]. These research consider authors with specific number of publications as experts and the number of co-authored publication as the weight of the generated graph.

One challenge with using DBLP though is the lack of important information such as keywords, abstracts and references of publications. Considering that keywords often reflect the subject of publications, they can be employed to serve as the expertise (skills) of the authors (experts). In addition, abstracts almost always provide a general overview of the subject. Adopting semantic analysis and topic mining techniques, one can mine keywords into the abstracts and determine the topic (area of expertise) of an author and his or her publications. Finally, number of references to a publication determines the influence the publication has in a particular research area. This influence then can be used to generate a general rating value (expert cost) and measure the impact of the authors in a particular research area (expertise level).

Having these pieces of information along with the results of analysis in hand, I was able to constructed collaboration between experts (relationships). Then I computed the scale of each collaboration given each experts skill (Communication Cost). Furthermore, I calculated the nec-

[1] www.linkedin.com
[2] www.informatik.uni-tier.de
[3] www.imdb.com/interfaces

essary relationship strength between a set of experts (Chemistry Level) to perform a task as a team. Having quantified the parameters successfully, I implemented ChemoTF and conducted the experiment.

To this end the purpose of my corpus is to provide an understanding of social ties between experts in a meaningful and measurable fashion. Furthermore, my corpus gives a new impetus to multi-facet chemistry-oriented relationships between experts by providing skill-set and area of expertise of each expert and the level of collaboration between them. This opens a new domain for further research on the area of team formation in social networks. Given the information my corpus provides and the purpose it serves for, according to Kennedy's corpora classification in [80], CompScholarCorp falls into *specialised corpora* category. A specialised corpus contains text of a certain type and aims to be representative of a certain language in order to answer very specific research questions. To answer these questions, CompScholarCorp accommodates information from three main sources that are fully elaborated in the next section.

4.3 Data Sources

Three different data-source have been used to construct CompScholarCorp. These include DBLP, Arnetminer citation network [122] and data collected from the web using my own scraper. The next three subsidiary sections describe these data-sources.

4.3.1 DBLP

I obtained a snapshot of DBLP on March 10, 2016 in XML format. This version of DBLP includes 2,983,857 nodes representing publications in the field of computer science along with 1,420,341 nodes representing authors. Each publication node contains meta-data consisting of the date and a key referring to the author's node. Publications nodes are tagged

```
<article mdate="2013-11-04" key="journal/corr/NajaflouJXY013">
   <author>Yashar Najaflou</author>
   <author>Behrouz Jedari</author>
   <author>Feng Xia</author>
   <author>Laurance T.Yang</author>
   <author>Mohammad S.Obaidat</author>
   <title>Safety Challenges and Solutions in Mobile Social Networks</title>
   <pages>834-854</pages>
   <year>2013</year>
   <volume>9</volume>
   <journal>JSYST</journal>
   <number>18</number>
   <url>db/journals/acta/corr9.html#NajaflouJXY013</url>
   <ee>http://doi.org/10.1109/JSYST.2013.2284696</ee>
</article>
```

Figure 4.1: A sample publication [102] as presented in DBLP dataset

by their types. The types of publication in this dataset include journal papers, books and thesis, book sections or collections, and conference proceedings. Apart from the types of the publications, DBLP provides information such as authors' full name tagged as *author*, publication title tagged as *title*, *page*, *year*, *volume*, *publisher*, name of the journal if applicable tagged as *journal*, link to DBLP online record tagged as *url*, and link to publication's page on publisher's website tagged as *ee*. Figure 4.1 demonstrates a sample publication node in DBLP.

DBLP provides a good source of data for publications. However it lacks some key pieces of information for the purpose of my thesis. Apart from the unavailability of keywords and abstracts in DBLP dataset, one important data field for my research is citation information. Although in some rare cases references to a few publications have been provided and tagged as *crossref*, this information is very limited and missing for majority of the publications. For this reason, I obtained citation information from another dataset called Arnetminer citation network. Combining this dataset with DBLP produced much more desirable results.

4.3.2 Arnetminer

Arnetminer [4] is a free bibliographic online service which integrates academic publication data into a social network in order to provide search and mining capability for social network analysis. It also provides a citation dataset which consists of connections between researchers, conferences, and publications extracted from DBLP and ACM.

I obtained the 8th version of this dataset on August 10, 2016 in TXT format. This version contains 3,272,991 publications and 8,466,859 citations dated up to July 17, 2016. Publications are presented within blocks using special indicators. Each line starts with a specific prefix indicating an attribute of the paper. These attributes include publication *title* marked by *#**, *authors* marked by *#@*, year marked by *#t*, *venue* marked by *#c*, *index id of the publication* marked by *#index*, *index id of references to the paper* (if any exists) marked by *#%* , and in some rare cases *abstract* marked by *#!*.

Having the citation information in hand, I combined them with DBLP and generated a dataset. Since information regarding abstracts and keywords were available in both of the datasets, I decided to extract publications links from *ee* nodes of DBLP and collected the rest of the required data from the web.

4.3.3 Web Data

As discussed earlier in this chapter, one important objective of my corpus was to include keywords and abstracts of publication. However, none of the two previous datasets provide this information. In order to achieve such an objective, I used *ee* child nodes of publications nodes in DBLP to access publications online pages. Then I retrieved keywords and abstracts by devising an automated scraper. The reason for employing a scraping technique was that I could not obtain these information by for-

[4] www.Arnetminer.org

mal ways. I first contacted Victoria University of Wellington library asking if they could provide this information from their digital resources. However I was advised that the library did not keep any track record of external publications as they were provided by the publishers. Then I looked at the possibility of using publishers' APIs through which keywords and abstracts could be acquired. I first contacted the largest digital libraries namely IEEEXplore [5], ACM [6], SpringerLink [7], and ScienceDirect [8]. Unfortunately they advised that they could not provide an API for this purpose. I also contacted DBLP and was given an API which did not provide any information regarding keywords and abstract.

With no reliable ready-made digital information at hand, my only option was to obtain required data from the publishers' web-pages. For this purpose, I extracted *ee* nodes from DBLP and used them as the starting point for my data collection process. Depending on the type of publication and publisher, *ee* nodes either contain a direct URL or a URL which redirects to publishers web-page. The web-pages are hosted by publishers and usually contain key information about publications. As keywords and abstract are usually considered key information for a publication, in most cases they are publicly and freely accessible in these pages. Moreover, some publications have more that one *ee* node. This is mainly because the work might have been published in more than one online page hence it has more than one link. On the other hand, in some other cases no *ee* node has been provided inside the publication node hence these publications could not be associated with any online links.

Having extracted URLs from DBLP and observed the web pages, it became evident that the pages do not reflect similar patterns or templates to present information. This is because each publisher prefers a different style of information presentation. For example, some publishers tend to

[5] www.ieeexplore.ieee.org
[6] www.dl.acm.org
[7] www.link.springer.com
[8] www.sciencedirect.com

publish the papers in full text, whereas some others prefer publishing abstracts only. Other examples are the difference in the page extension (i.e. various extensions such as PDF, PS, XML, HTML, etc.), difference in the markup (i.e. various html tags), difference in page templates or layout (i.e. frames and JQuery objects), and different places on the page to publish information. This wide variety of information presentation could make the data collection process extremely difficult and the collected data undesirable. A common practice among researchers to overcome this difficulty is to ignore the unnecessary information, often called noise [38], by studying the different forms of data presentation and producing various sets of criteria by which the captured data can be purified. The next section elaborates what criteria was specified for my corpus, what data collection strategy was adopted, and how the intended data was captured.

4.4 Sampling and Data Collection

I started the process of collecting data by defining a set of criteria for downsizing the information obtained from the web. All the information collected for the corpus is publicly available on the publisher's websites with no sign-in required for access. In the beginning, I extracted the URLs from DBLP and investigated the web pages. Having observed the web paged carefully, it became evident that in most cases the information presentation varies according to the publisher. In other words, each publisher uses a single template to present its publications. It was also found that in some cases this presentation varies according to the type of publication as well (*i.e.* SpringerLink uses different templates for journal articles, conference proceedings, and books).

In order to address this issue, I studied the URLs and found a connection between the URL of each publication and its corresponding web page on publisher's website. This task was rather complex as it required classifying all the URLs by common domains or hosts while URLs were

not indicative enough of the domains they referred to. Redirect URLs often contain the word "DOI" in their name rather than the name of the original domain. For an instance, the URL for the sample publication demonstrated in Figure 4.1 redirects to a web page on IEEE domain [9] where the article [102] has been published. For this reason, I identified the DOIs, specified the domains they belong to, and determined the types of publication they refer to.

Being able to identify the source of each URL, I classified the URLs into 345 style/template categories by which the desired information was presented on the web pages. This was an extremely time-consuming but very vital task for constructing my corpus. Samples of URLs referring to identical web resources were carefully chosen and studied. Then they were put into categories and were tested to ensure the produced results are both desirable and reliable. This step was important because the categories determined the regulations of data collection. Without this categorisation, one would have to define a single rule to contain all types of presentations which would be an extremely difficult task.

Once I defined the style templates, I was able to perform data collection. To achieve this, I adopted a technique often referred to as *scraping* or *crawling*. The aim was to automatically parse the content of HTML pages derived from URLs and extract abstracts and keywords according to the category of each URL. An scraper is a virtual automated browser which understands only certain tags it is built to capture using HTTP requests [95]. Given the large number of these URLs in my case (6,452,042 *ee* nodes in total), it would be extremely hard to obtain the desired information manually without the use of a scraper. One would need to open the pages one by one, look for keywords and abstract on the page, and copy and paste the content individually.

The motivation behind scraping was mainly due to the unavailability of the required data through publishers' APIs. In addition, the various

[9] http://ieeexplore.ieee.org/document/6642041/

patterns, templates and technologies employed by publishers to present the data made it almost impossible to use generic software for collecting the data. Scraping large amounts of data requires a careful consideration with respect to design. The more the number of web requests gets, the more resources are required and the more difficult managing the process becomes. As explained earlier in Section 4.3.3 of this chapter, it is quite common for a publication to have more than one web address. This is because a single paper can be published by various publishers. This certainly challenges the scalability of the architecture as it can affect the process speed, time management, resource management, and information extraction in general. To tackle this issue, I designed and implemented an engine to process the web requests in a timely, manageable, and recoverable fashion.

Scraping large amounts of data also brings the challenge of dealing with large number of hosts. Each host has its own particular limitations for *HTTP* requests. Some allow a particular amount of time between the requests whereas some require *cookies* in order to provide service. In addition, some hosts require a certificate agreement and some others use tools such as *JavaScript*, *JQuery* or *Frames*. As shown in Figure 4.2, illustrating the mark-up used to present the previously given sample article in Figure 4.1, data regarding the abstract and keywords are often wrapped inside a *JavaScript* which makes the retrieval of the information complex. Moreover, information is stored and marked up differently in each individual website. In order to overcome this challenge, I equipped my scraper with a set of accessories and devised a mechanism by which the scraping process adjusts according to the specifications of each of the URL categories.

The scraper I designed and implemented takes advantage of the architecture illustrated in Figure 4.3. As observed from this figure, the URLs of all publications are first obtained from DBLP. These URLs are put into a special structure which contains information such as the identifier of

```
<script type="text/javascript">
var body_rightsLink ="", body_publisher = "";
var recordId = "";

var global = { document: { disqus:{
            remote_auth_s3 : '',
            public_api_key:'1lxKgMbpNIbQvfk2tqLcWeSVloE8rgIY2CV1tCu3Vp641oL4eEITYE
            short_name:'ieeexplore',
            client_url:'http://ieeexplore.ieee.org',
            sso_enabled:'{$sessionProfile.provisioned}'},
    fullTextAccess: false}
};
global.document.metadata = {
    "userInfo": {
        "institute": false, "member": false, "individual": false, "guest": false,
    }, "authors": [
        { "name": "Yashar Najaflou", "affiliation": "Sch. of Software, Dalian Univ
        { "name": "Behrouz Jedari", "affiliation": "Sch. of Software, Dalian Univ.
        { "name": "Feng Xia", "affiliation": "Sch. of Software, Dalian Univ. of Te
        { "name": "Laurence T. Yang", "affiliation": "Sch. of Comput. Sci. &amp;
        { "name": "Mohammad S. Obaidat", "affiliation": "Dept. of Comput. Sci. &ar
            "citationCountPaper": 14, "citationCountPatent": 0, "totalDownloads":
        }, "sponsors": [{ "name": "IEEE Systems Council", "url": "http://www.ieees
           "displayPublicationTitle": "IEEE Systems Journal",
    "keywords": [
        { "type": "IEEE Keywords", "kwd": ["Safety", "Social network services", "S
        { "type": "INSPEC: Controlled Indexing", "kwd": ["delay tolerant networks"
        { "type": "INSPEC: Non-Controlled Indexing", "kwd": ["safety challenges",
        { "type": "Author Keywords ", "kwd": ["Mobile social networks (MSNs)", "op
        "title": "Safety Challenges and Solutions in Mobile Social Networks",
        "abstract": "Mobile social networks (MSNs) are specific types of social me
};
</script>
```


Figure 4.2: A snapshot of a sample publication [102] presented by IEEE in HTML mark-up, highlighting keywords and abstract information.

the publication and authors so they can be used later while creating corpus. For simplicity, from now on, I refer to this structure as URL throughout the thesis. They are then sent to *Determine Category* function which classifies each URL according to the criteria I developed earlier. Then it produces threads for each category and sends them to *Download Engine* function. Each thread carries a list of URLs of a particular category and is processed independently. The main responsibility of the engine is to create *HTTP GET* requests for every URL and to ensure each URL is processed successfully and ends in a desirable result.

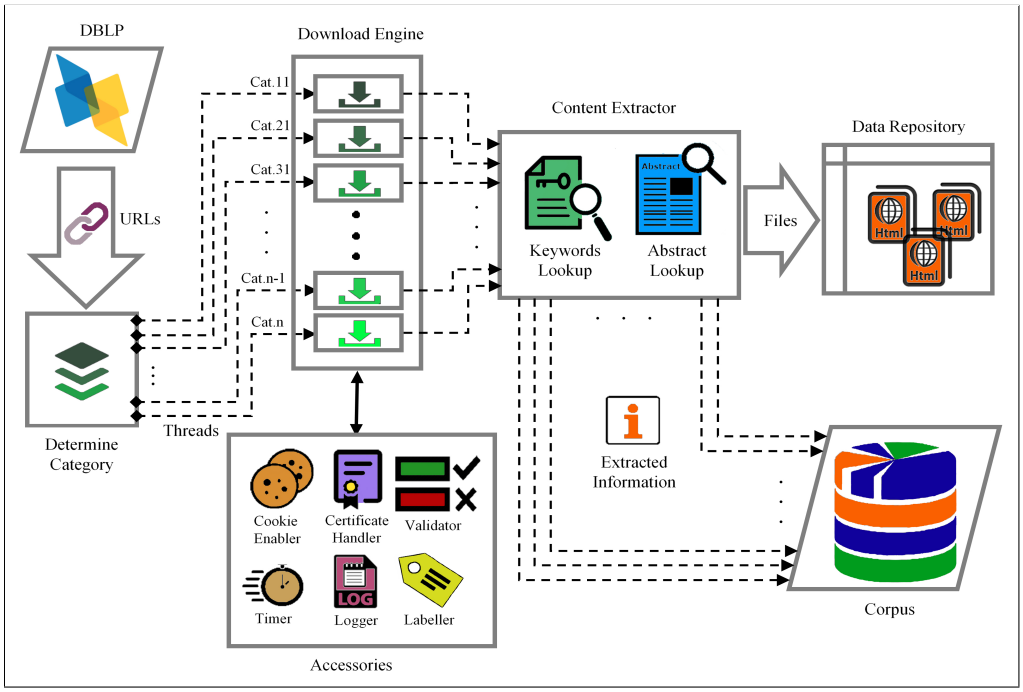


Figure 4.3: The architecture of my scraper for collecting required data

A successful process in the engine has the characteristics of being reliable, timely, independent, documentable, and recoverable. This is the main intuition behind accessories. *Cookie Enabler* enables retrieving information from the hosts that require cookies. *Certificate Handler* catches the invalid certificate exceptions thrown by some websites and handles them by ignoring *SSL Trust* request. *Timer* makes sure there is enough waiting time between each process. This enables the engine to produce web requests in appropriate time sequences which prevents the scraper from overloading a website by multiple requests. The generated waiting time is random to avoid consequent blocking, service suspension, and/or interruptions by hosts. *Logger* records every event and its corresponding responses. *Validator* checks to see whether the files downloaded files are valid and contain desirable information. Furthermore, if a publication has more than one URL, it checks them all and puts together the best pos-

sible result. Finally the *Labeller*, names the downloaded files so that they can be accessed from *Data Repository*, if required, and can be associated with their related records in the corpus.

After the validation and labelling process are completed in the engine, the downloaded HTML files are delivered to *Content Extractor* function in their specified threads and defined category. This function plays the role of a *spider* in a traditional scraping system which is to look for particular information inside the HTML mark-up. The function contains all the category-oriented rules and conditions to extract keywords and abstracts. As each thread carries only a specific category of downloaded data, this function can apply the suitable rules according to the specified categories of each downloaded file.

The downloaded files are then archived in *Data Repository* for further use if necessary. Their names which have been produced and labelled by *Labeller* in the engine, distinguishes their corresponding URL in DBLP and the publication node in the CompScholarCorp. Finally the process ends by handing over the extracted information to a simple XML writer which records the information in the corpus in the specified format designed and implemented in Section 4.5 of this chapter.

Although my scraper captured abstracts and keywords from a large number of publications, some web pages could not get parsed. These exceptions mainly include those publications that their corresponding web page was in non-HTML format or did not follow any consistent mark-up. To obtain any information from these pages requires a manual information retrieval process. Given the relatively large number of these type of publications, it was decided to remove them from the corpus. Another sets of example include publications for which no such information was available on the web pages or the pages did not exist at all.

Having extracted abstracts and keywords from the web pages, I integrated the obtained data with data gained from DBLP and Arnetminer datasets into one single corpus. I first identified the required information

in the two datasets. Given that abstract and keywords for certain publications were not available or could not be retrieved, I only kept the publications with valid abstract and keywords. In addition, instead of publication references, citations to publication are derived from Arnetminer dataset and placed into each publication node so that no further process are required to extract citations. Since both DBLP and Arnetminer use DBLP-key to identify their records, future research can benefit from latest updates released by these source and match them using this key.

The information each publication reflects in my corpus includes *type* of the publication, publication *date*, *authors' full name*, *DBLP-key* for authors, *title* of the publication, publication *venue*, *urls* to publication web pages, *citations* to the publications, and finally *abstracts* and *keywords*. The comprehensive information this corpus provides makes it an asset for future research. In order to facilitate the data retrieval process from this corpus, I designed a structure and formatted the data presentation using XML markup. The next section describes the decisions made during the process and elaborates the design.

4.5 Structure Design

One of the challenges of constructing a corpus is designing the structure. This is because the structure of a corpus should be ideally designed in a reliable, flexible, reusable, extensible, and portable fashion. In order to achieve such a design, I adopted the popular eXtensible Markup Language (XML) structure and defined new identifiers to presented the corpus information while keeping the corpus compatible with DBLP.

XML is a markup language which describes data in a customisable tag based format. The flexibility involved in the design of XML documents has made it a standard for transmitting data on the web. [19]. I used XML for the structure design because it guarantees the reliability, extensibility and portability of the corpus. It is platform-independent and is widely

used across database engines. In addition, feeding and retrieving information to and from XML is rapid thanks to the modifiable and extensible mark-up system it provides.

In order to ensure that CompScholarCorp is reusable and compatible with prior research, I have kept DBLP-key information. Furthermore, to ease information retrieval process, I introduced new identifiers which are unique to CompScholarCorp. For example the author are divided into multiple nodes, each referring to one single author node with a designated key. Another example is the publication type which has a designated node in my corpus. Figure 4.4 exhibits the XML schema designed for CompScholarCorp. In order to provide a better understanding of the structure of CompScholarCorp and the information it reflects, Figure 4.5 presents a sample publication node from this corpus. This is the same article presented in DBLP illustrated in Figure 4.1 and its markup on IEEE website was depicted in Figure 4.2. As observed from comparing these three records, my structure for the corpus provides a cleaner more consistent structure as a source for information retrieval and for analytical purposes. Furthermore, since both DBLP and Arnetminer use DBLP-key to identify their records, future research can benefit from latest updates released by these source and match them using this key.

To this end, this chapter discussed how required data from the publications in computer science was collected and how an expertise corpus called CompScholarCorp was designed and compiled. The process of data collection, sampling, design and construction of this corpus was thoroughly explained and examples of the records in this corpus were outlined. In order to provide a broader understanding of the information this corpus contains, the following chapter visualises CompScholarCorp in a series of illustrations, graphs and statistical tables. The visualisation is done from the perspective of different entities of this corpus in order to map them with ChemoTF concepts introduced earlier in Chapter 3.

```
<?xml version="1.0" encoding="UTF-8" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
   <xs:element name="publication">
      <xs:complexType>
         <xs:sequence>
            <xs:element name="title" type="xs:string" maxOccurs="1"/>
            <xs:element name="type" type="xs:string" maxOccurs="1"/>
            <xs:element name="date" type="xs:string" maxOccurs="1"/>
            <xs:element name="venue" type="xs:string" maxOccurs="1"/>
            <xs:element name="authors" maxOccurs="1"/>
               <xs:complexType>
                  <xs:sequence>
                     <xs:element name="author" minOccurs="1">
                        <xs:complexType>
                           <xs:attribute name="key" type="xs:IDREF"/>
                        </xs:complexType>
                     </xs:element>
                  </xs:sequence>
               </xs:complexType>
            </xs:element>
            <xs:element name="abstract" type="xs:string" maxOccurs="1"/>
            <xs:element name="keywords" maxOccurs="1"/>
               <xs:complexType>
                  <xs:sequence>
                     <xs:element name="keyword" minOccurs="1">
                        <xs:complexType>
                           <xs:attribute name="key" type="xs:IDREF"/>
                        </xs:complexType>
                     </xs:element>
                  </xs:sequence>
               </xs:complexType>
            </xs:element>
            <xs:element name="citations" maxOccurs="1"/>
               <xs:complexType>
                  <xs:sequence>
                     <xs:element name="citation">
                        <xs:complexType>
                           <xs:attribute name="key" type="xs:IDREF"/>
                        </xs:complexType>
                     </xs:element>
                  </xs:sequence>
               </xs:complexType>
            </xs:element>
         </xs:sequence>
         <xs:attribute name="key" maxOccurs="1"/>
      </xs:complexType>
   </xs:element>
</xs:schema>
```

Figure 4.4: CompScholarCorp XML schema

```
<publication key="journal/corr/NajaflouJXY013">
   <title>Safety Challenges and Solutions in Mobile Social Networks</title>
   <type>Journal</type>
   <date>2013-11-04</date>
   <venue>IEEE_JSYST</venue>
   <link>http://doi.org/10.1109/JSYST.2013.2284696</link>
   <authors>
      <author key="54134">Yashar Najaflou</author>
      <author key="52198">Behrouz Jedari</author>
      <author key="1021">Feng Xia</author>
      <author key="8134">Laurance T.Yang</author>
      <author key="110">Mohammad S. Obaidat</author>
   </authors>
   <abstract>Mobile social networks (MSNs) are specific types of social media
         which consolidate the ability of omnipresent connection for mobile
         users/devices to share user-centric data objects among interested users.
         Taking advantage of the characteristics of both social networks and
         opportunistic networks (OppNets), MSNs are capable of providing an
         efficient and effective mobile environment for users to access, share,
         and distribute data. However, the lack of a protective infrastructure in
         these networks has turned them to convenient targets for various perils.
         This is the main impulse why MSNs carry disparate and intricate safety
         concerns and embrace divergent safety challenging problems. In this
         paper, we aim to provide a clear categorization on safety challenges and
         a deep exploration over some recent solutions in MSNs. This work narrows
         the safety challenges and solution techniques down from OppNets and
         delay-tolerant networks to MSNs with the hope of covering the work
         proposed around security, privacy, and trust in MSNs.
   </abstract>
   <keywords>
      <keyword key="1021">Mobile</keyword>
      <keyword key="5741">Social Network</keyword>
      <keyword key="89941">MSN</keyword>
      <keyword key="97974">Security</keyword>
      <keyword key="621">Privacy</keyword>
      <keyword key="2281">Trust</keyword>
   </keywords>
   </citations>
      <citation key="journals/tnsm/ZhangCW16"/>
      <citation key="journals/cn/MousaMHYHB15"/>
      <citation key="journals/cas/RezvanianM15"/>
      <citation key="journals/scn/ZhangWLLC16"/>
      <citation key="journals/access/AbbasRO16"/>
      <citation key="journals/access/WuYYZQHH16"/>
      <citation key="conf/asunam/FanFLY14"/>
      <citation key="conf/ccgrid/AbbasRWEO16"/>
   </citations>
</publication>
```

Figure 4.5: An example of a publication in CompScholarCorp

> "We must design the message in a way that leads readers on a journey of discovery, making sure that what's important is clearly seen and understood. Numbers have an important story to tell. They rely on you to give them a clear and convincing voice."
>
> *Stephen Few* quoted in [1]

Chapter 5

Data Visualisation

The corpus compiled in the previous chapter contains a large amount of data with unique characteristics. The information this rich corpus provides makes it a valuable asset for research in various fields including the field of computer science and in the area of TF. In order to explore the potential of this corpus, this chapter visualises it in a series of illustrations, graphs and statistical tables. The aim of this visualisation is to ease the understanding of the concepts introduced in this thesis and to identify the possibilities that CompScholarCorp can be further used for.

The first section provides an overview of CompScholarCorp and its various entities. The characteristics of each entity is outlined and briefly discussed. In addition, some general statistics are presented. The second section visualises the corpus from the perspective of skills and presents this in the form of tables, figures and network. The third section visualises the corpus from the perspective of authors and describes it according to their characteristics. Furthermore statistics regarding the top authors based on the number of publication and citations are given. The final section visualises the networks derived from CompScholarCorp. The collaboration network is explored, then the citation network is scrutinised and relative visualisations are illustrated.

5.1 CompScholarCorp Overview

CompScholarCorp encompasses a big body of publications in the field of computer science. It contains 1,044,454 publications including 21,108 book chapters (books, book chapters, thesis and collections), 570,101 conferences (conference proceedings and workshop papers), 48,209 editorships (editorship and informal publications), and 405,036 journal papers. These publications relate to the period 1974-2016 (March 10 2016). Figure 5.1 demonstrates the number of publications included in CompScholarCorp in each year with the publication type.

As observed from Figure 5.1, conference papers have the highest occurrence rate followed by journal articles, book chapters and editorships respectively in this corpus. The reason for this is that the number of conferences held in computer science are generally very high thus conferences proceedings outnumber other types of publications. Moreover, the number of publications has increased over time and more recent years exhibit more publications when compared with previous years. One possible reason for this is the popularity that the field of computer science has been gaining over the years among researchers. The advent of new technology and its influence on interdisciplinary fields are the main drive for this popularity. The linear rise of the the number of publications over the years suggest that this rise will continue to appear in the future.

In addition to the date and type, publications in CompScholarCorp contain other useful information such as *author(s)*, *title*, *venue*, *DBLP-key*, *URL* to publication web pages, *citations*, and finally *abstracts* and *keywords*. To give an overview of my corpus, Table 5.1 summarises the number of key entities including keywords, authors, publications, and citations and provide statistical measures including highest, lowest, average and median values. There are 472,365 unique authors where each are in collaboration with at least one other author. In addition, there are 24,500 unique keywords which represent publications as well as the au-

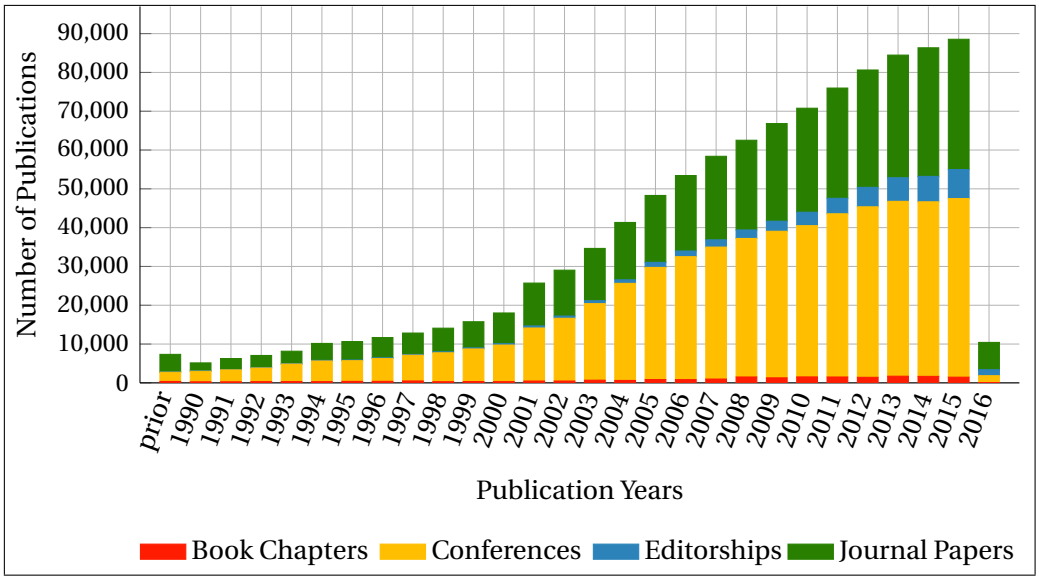


Figure 5.1: Number of Publications per year/type in CompScholarCorp

thors. The overall number of citations in CompScholarCorp is 3,466,859.

It is important to emphasise that no corpus is complete. It is because a corpus is a representative samples which provides an opportunity to test and challenge ideas and intuitions about the information it represents, as apposed to digital libraries, datasets, and online indexing services that focus on storing and serving up-to-date all-inclusive data. The main objective of CompScholarCorp is to provide data for studying the characteristics of computer science research communities such as, trends, behaviour, scale of collaboration, scale of influence, and underlying relationships. Albeit being comprehensive, it mirrors a snapshot of research communities and has been compiled to serve as a testbed for ChemoTF experiment. The is the main reason why the number of publications, authors and citations in CompScholarCorp is lower compared to well-known digital bibliographical services such as, *Google Scholar*[1], *Aminer*[2],

[1] www.scholar.google.com
[2] www.aminer.org

Table 5.1: Overview of CompScholarCorp with respect to various entities

Value	Keyword		Author		Publication		Citation	
Per	*Pub.*	*Auth.*	*Pub.*	*Keyw.*	*Cit.*	*Auth.*	*Pub.*	*Auth.*
Highest	15	89	16	48,203	181	1,290	21,008	26,221
Lowest	2	2	2	31	0	1	0	0
Average	6.13	20.46	5.13	1,922.1	19.20	14.23	13.18	16.94
Median	5	11	4	1,081	12	8	7	8
Total	24,500		472,365		1,044,454		3,466,859	

and SemanticScholar[3]. In addition, some publication types have been excluded from CompScholarCorp which reduces the overall number of publications as well as citations and authors. These publication types include technical reports, book reviews, dictionaries, encyclopedias, newspaper and magazine articles, textbooks and posters.

In order to provide a more detailed look over the corpus, the rest of this chapter has been dedicated to inspecting CompScholarCorp from the perspective of the key entities demonstrated in Table 5.1. First the keywords of the publications are explored and related data is visualised. Then the authors are investigated and some statistics are provided. Finally the networks drawn from the corpus including collaboration and citation networks are visualised and discussed.

5.2 Keywords: Skills

Keywords are the unique entities of CompScholarCorp which distinguish this corpus from corpora with similar purposes. As discussed earlier, the keywords of each publication are gathered from the keywords appearing

[3]www.semanticscholar.org

in the scientific article. The assumption is that these keywords represent the key concepts of a research or the methods used in a study. From this perspective, keywords can be considered skills required to produce the corresponding work.

As demonstrated earlier in Table 5.1, 24,500 unique keywords have been identified in CompScholarCorp. The highest number of keywords appearing in a publication is 15 whereas the lowest number of keywords in a publication is 2. Moreover, the average and median value of keywords throughout the corpus are 6.13 and 5 respectively. These values for keywords per authors are 89 as the highest, 2 as the lowest, 20.46 as the average, and 11 as the median value. Figure 5.2 illustrates keywords in this corpus as a text cloud. The larger the size of the font is, the more frequently the keyword has appeared in the corpus.

Keywords can be utilised for a variety of purposes. One application which has been introduced in this thesis is *skill cost*. Considering keywords as skills and authors as experts, the cost of a skill is calculated as the average cost of experts who possess the skill. As described in Chapter 3, ChemoTF uses skill cost as a budget constraint to select financially affordable experts. Table 5.2 demonstrates top 20 keywords with the highest number of frequency and their cost.

Another purpose keywords can be utilised for is to determine the relationships between skills and analyse the corpus accordingly. The intuition behind such analysis is that there are research areas for which some keywords commonly appear together. Identifying these areas and the common keywords for each area provides an understanding of necessary skills and the scale of their influence in each area of expertise.

Figure 5.3 visualises the collaboration network derived from CompScholarCorp and highlights the most frequently used keywords using the *Fruchterman-Reingold* method [56]. In this network, vertices represent authors possessing keywords and edges represent the collaboration between them. This method is based on a *Force-Directed* algorithm in which

Table 5.2: Top 20 Most Frequently Used Keywords in CompScholarCorp

#	Keyword	Freq.	Cost	#	Keyword	Freq.	Cost
1	InformationRetrieval	67,660	2.81	11	Learning	11,613	4.03
2	SearchEngine	33,327	3.26	12	MachineLearning	9,100	3.99
3	WWW	31,112	3.29	13	Probability	8,331	3.98
4	SocialNetwork	25,439	3.38	14	Ranking	7,977	4.09
5	Database	21,893	3.41	15	TextCorpus	7,829	3.18
6	Statistics	15,334	3.54	16	Prediction	7,829	4.96
7	Computing	14,152	3.71	17	Ontology	7,445	3.24
8	ClusterAnalysis	13,148	3.70	18	Recommender	7,416	4.98
9	Semantics	12,468	3.82	19	LanguageModels	6,766	4.84
10	Software	12,291	3.97	20	Algorithm	6,706	5.10

Figure 5.2: A text cloud highlighting keywords in CompScholarCorp. The larger the font size is, the higher the frequently of keyword is.

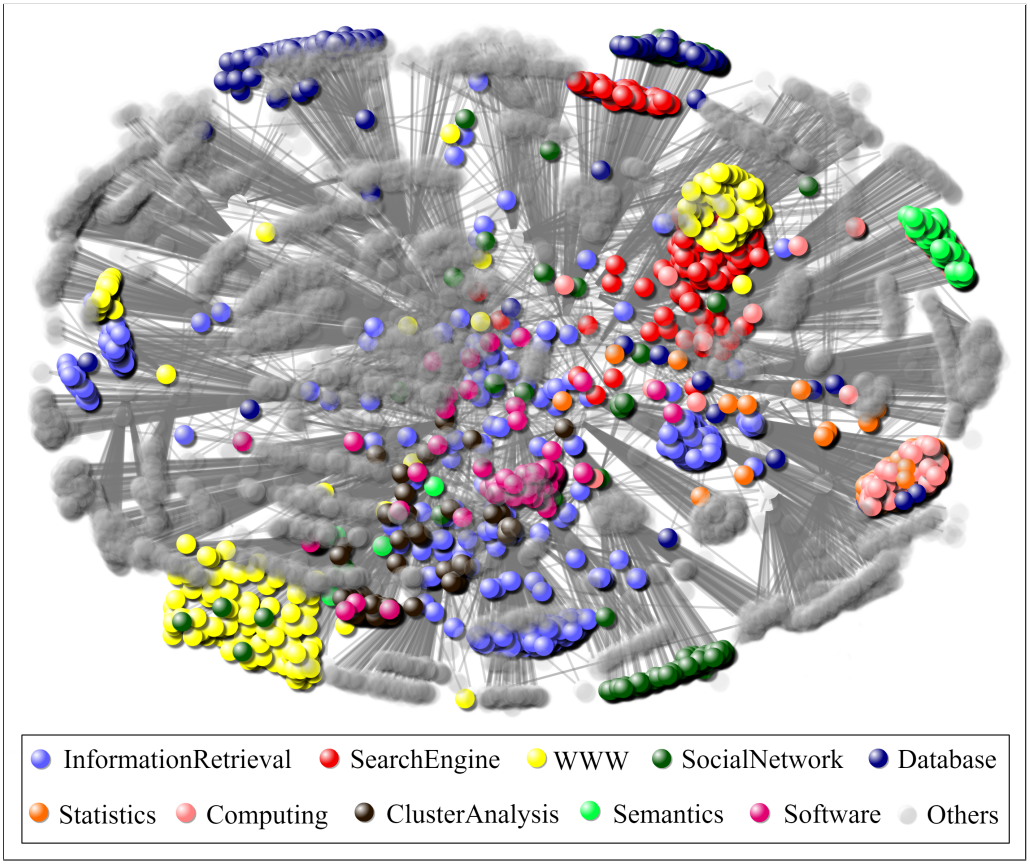


Figure 5.3: Visualisation of CompScholarCorp collaboration network using Fruchterman-Reingold method [56] highlighting top 10 keywords

the vertex layout is determined by the forces pulling vertices together and pushing them apart. Attractive forces occur between adjacent vertices only, whereas repulsive forces occur between every pair of vertices. Each iteration computes the sum of the forces on each vertex, then moves the vertices to their new positions.

Figure 5.3 shows there are close bonds between some highlighted keywords. For example the bonds between *SearchEngine* (red) and *WWW* (yellow), *Computing* (light pink) and *Statistics* (orange), and *WWW* (yellow) and *InformationRetrieval* (light blue) are noticeable. 78.2% of the papers that have used *SearchEngine* as their keyword, have also used *WWW*.

The rate for papers with *Computing* and *Statistics* is 53.5%, and for papers with *WWW* and *InformationRetrieval* is 38.6%. This suggests a chemistry between various keywords which might lie in the nature of the concepts these keywords create together. For an instance, the concept of *Online Search Engines* demands knowledge of both *SearchEngine* and *WWW*. Furthermore, I found that this chemistry is asymmetric – meaning that the reverse ratio does not necessarily apply if the propositions are reversed. For example, only 11.4% of the papers with *WWW* keyword accommodate *SearchEngine*, as opposed to the ratio 78.2% discussed earlier for these keywords – of course this because there are other bonds with other less frequently used keywords that have not been highlighted in my visualisation.

Another noticeable point is the heterogeneous nature of some keywords that are distributed across the network and are coupled with other keywords. Some good examples of these keywords are *InformationRetrieval* (light blue) and *Software* (dark pink) which both have bonds with multiple keywords. This suggests that they are necessary part of multiple areas of expertise and can be associated with various fields. I found that 5.3% of all the keywords have bonds with *InformationRetrieval*. This ratio is 2.1% for *Software*.

Having visualised the corpus from the perspective of keywords, the following section visualises it from the perspective of authors.

5.3 Authors: Experts

Authors are important entities of CompScholarCorp. They possess different skill-sets and together they construct the rich multi-skilled society. TF algorithms which use bibliographical information as their dataset consider authors as experts and the society of authors as a pool of experts where the best team is selected from based on criteria. In this respect, ChemoTF also consider authors as experts.

CompScholarCorp encompasses all the authors in the field of Computer Science regardless of their number of publications or level of expertise. However as the aim of ChemoTF is to form teams with the highest level of expertise possible, I acknowledge top 10 authors with the highest number of publications in CompScholarCorp in Table 5.3. The table also reveals the highest Expertise Level of each author given his or her most frequently used keyword as well as some largely used keywords. As stated in Chapter 3, the Expertise Level is calculated as the sum of experiences associated with a particular skill. In the context of CompScholarCorp, Expertise Level of an author in a keyword refers to the sum of the number of the publications of the author where the particular keyword appears.

In addition to the number of publications, authors are also associated with the number of citations to their publications. The number of citations to a publication reveals how much the publication has been endorsed in the scientific community. In this thesis, citations are used to calculate the Expert Cost of authors which is computed based on the *h-index* [68] of authors. The logic behind using h-index as Expert Cost is that this metric considers both the number of publications and the number of citations. Table 5.4 demonstrates top 10 authors with the highest number of citations in CompScholarCorp along with their Expert Cost.

It is crucial to understand that neither the number of publications nor the number of citations are indications for the quality of authors. It is highly debatable whether numerical ranking of authors leads to fair comparisons. The reason for such dispute lies in four main reasons including the unclear representation of authors, various behaviour patterns of publications in different research areas, the subjective nature of such comparison, and inability to cover all the publications of all authors.

The first reason for discouraging quantitative comparison between authors is that numerical reasoning fails to provide a complete image of the quality of a publication or an author. There are prolific authors with extraordinarily high impact on research communities but with relatively

Table 5.3: Top 10 authors with the highest number of publications in CompScholarCorp with their highest Expertise Level and keywords

Rank	Author	# Pub.	Highest ExpLvl | Top Keyword	Other Keywords
1	H. V. Poor	1,290	371 | Information Theory	Signal Processing, Decoding, Game Theory
2	P. S. Yu	910	491 | Data Mining	Database, Indexing, Social Network, Internet
3	W. Gao	891	210 | Video Coding	Face Recognition, Decoding, Feature Extraction, Encoding
4	Yang Liu	885	387 | Artificial Intelligence	Algorithms, Neural Networks, Image Processing, Learning
5	M. Alouini	870	475 | Signal Processing	Fading, Bit Error Rate, Fading Channel, Cognitive Radio
6	Y. Zhang	852	292 | Nanoparticles	Second Order, Nonlinear Optics, Imaging, Image Processing
7	J. Li	833	301 | Wireless Networks	Sensor Networks, Optimization, Distributed Systems
8	L. Hanzo	829	265 | Bit Error Rate	Decoding, Signal Processing, Fading, Channel Coding
9	L. Zhang	796	251 | Numerical Simulation	Human Error, Apoptosis, Combustion, Analysis
10	J. Wang	788	355 | Neural Network	Genetic, Algorithms, Networks, Genomics

Table 5.4: Top 10 authors with the highest number of citations in CompScholarCorp with Expert Cost and most frequent keywords

Rank	Author	# Cit.	Expert Cost	Keywords
1	W. H. Press	26,221	3	Numerical Recipes, Computing, Monte Carlo
2	L. A. Zadeh	21,461	9	Fuzzy Logic, NLP, Soft Computing, Fuzzy Set
3	V. Vapnik	20,354	6	Machine Learning, Pattern Recognition, Vector
4	H. A. Simon	19,291	10	AI, Computer Simulation, Cognitive Processes
5	D. E. Goldberg	15,683	7	Genetic, Evolutionary Computation, Bayesian
6	G. Hinton	14,608	11	Neural Networks, Speech Recognition, Boltzmann
7	J. Han	12,710	13	Data Mining, Information Retrieval, Databases
8	R. Rivest	12,321	7	Cryptography, Public Key, Complexity
9	T. J. Sejnowski	12,105	12	Component Analysis, Oscillations, Simulation
10	A. Zisserman	11,889	11	Computer Vision, Pattern Recognition, Imagaging

small number of publications. A good example is the renowned scientist E. F. Codd, the inventor of the relational data models in [34], who has only 71 publications on CompScholarCorp but is acknowledged as an inspirational figure in field of Computer Science particularly in the area of Database. This leads to a belief that numerical ranking is not an appropriate method to draw conclusions.

The second reason to dismiss drawing judgements based on the number of publications or citations is that various sub-disciplines demonstrate a very different publication behaviour. For example authors in fundamental areas of Computer Science such as *Information Theory* or *Data Analysis* tend to have more publications thus higher number of citations compared to researchers in specific areas such as *Heuristic Algorithms*. The reason for a research area to be fundamental might be driven from its

historical background, practicality or popularity among the researchers. These are only few of the factors which needs to be considered during the comparison.

The third reason to support the discussion above is that comparing the quality of publications can be very subjective. Although CompScholarCorp encompasses comprehensive data regarding the types and venues of the publications, it is debatable whether this data can be used to assess the quality of a study. On one hand, there are inspirational conference papers which have been overwhelmingly embraced by scholars. On the other hand, there are journal articles which have not been acknowledged much. Even the reputability of venues is disputable.

The last reason to repudiate the idea of comparing authors based on numerical ranking is that CompScholarCorp includes only publications related to the field of computer science based on DBLP and Arnetminer dataset. Given that the coverage of various research areas in these two datasets is still quite inhomogeneous, some publications might have been unintentionally discarded. This may effect the variation of the total number publications for authors.

To this end, it is difficult to draw an objective conclusion and a fair comparison based on the number of the publications or Expertise level of the authors. Having visualised the corpus from keywords and authors perspective, the rest of this chapter describes the networks derived from CompScholarCorp and elucidates their characteristics.

5.4 Network of Experts

Data offered by CompScholarCorp in an organised fashion makes it possible to use this corpus for different types of analysis. One interesting type is to derive networks from this corpus and analyse them accordingly. For the purpose of this thesis, I constructed two networks including *Collaboration Network* and *Citation Network*. Collaboration Network was drawn

by connecting the authors based on their collaborations. This is the social network for which ChemoTF was implemented. The analysis run on this network leads to the calculation of the majority of ChemoTF parameters. Citation Network was constructed by connecting authors based on citations to their publications in Arnetminer dataset and the publication records in CompScholarCorp. This network is used to calculate the Expert Cost. The rest of this chapter visualises these two networks and provides a big picture of their attributes.

5.4.1 Collaboration Network

Collaboration network is a network of authors connected through their mutual publications. It is an undirected, unweighted looped multigraph and in my case it has $472,365$ vertices and $1,044,454$ edges in total. It is undirected because the collaboration relationship is symmetric. It is a looped multigraph as vertices represent authors rather than publications.

The first method I employ to visualise the collaboration network is *Barabási-Albert model* [13]. It is an iteration of *Power Law Degree Distribution* in which the distribution of edges to vertices (degree) are plotted on a log-log scale on which power law render as straight lines. This method assumes that the network is scale-free and the degree distribution over the vertices asymptotically follows a power law. The left graph in Figure 5.5 depicts this visualisation where y axis is the number of the vertices and x axis is devoted to degree function. The graph suggests that the degree distribution is a power law with exponent $\gamma = 3.12$ beginning at the minimum degree of $d_{min} = 51$. The actual minimum degree of this network is 1, whereas the maximum degree is $1,290$ and the median degree is 14.23. It is observable that as the degree increases for a vertex, the number of vertices possessing such degree become scarce. This behaviour suggests a negative correlation between degree and frequency of the vertices which I calculated at the rate -87%. Moreover, the graph gets

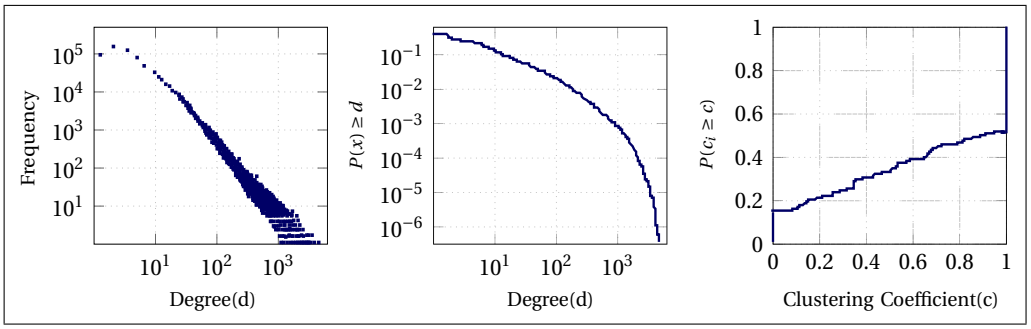


Figure 5.4: CompScholarCorp Collaboration Network Visualisation. From left to right: Power Law Degree Distribution, Cumulative Degree Distribution, and Local Clustering Coefficient

thicker as the degree increases. The reason for this behaviour is that the vertices possessing degree lower than average outnumber those vertices with higher than average degree. More specifically in this collaboration network, 89.9% of authors have fewer publications than the median average number of publications which is 8.

The degree distributions discussed above provide an overall picture of the network. However, it is often argued that many distributions are misidentified as power laws whereas in fact they are not [94]. In order to clarify this, I visualise the collaboration network using *Cumulative Degree Distribution* in a log-log scale on the middle graph in Figure 5.5. This visualisation indicates the probability of the degree of a randomly picked vertex being larger than the degree calculated by degree function. In other words, it demonstrates how often degree of a given size or larger occurs throughout the network. The y axis gives the probability of random degrees larger than the degree. This visualisation confirms that the degree distribution is a power law. The degree distribution function has a 96% goodness of fit with $y = 10^5 nx^{-m}$ while $3.69 \leq n \leq 5.70$ and $1.41 \leq m \leq 1.81$.

Furthermore, the right graph in Figure 5.5 visualises my corpus using *Local Clustering Coefficient*. It is a method to numerically measure the number of clusters in order to identify the extent to which edges in

a collaboration network tend to form triangles. In other words, the local clustering coefficient of a vertex is the probability of two randomly picked vertices of its adjacent vertices being connected. The y axis reveals the probability of adjacent vertices being connected while the x axis indicates the scale of clusters. A value of 1 on x axis, denotes that all possible triangles are formed, and value of zero suggests a triangle free cluster. It was found that 51% of the network is tightly clustered which signifies a rate of strong collaboration among communities. In addition 29% of the network is average to poorly clustered which specifies a weak level of collaboration between authors. The rest of the network is cluster-free which indicates the number of independent authors who have not collaborated. I calculated the clustering coefficient at 18% which indicates that 18% of the communities found in the network have authors who collaborate.

5.4.2 Citation Network

The second network of this thesis is Citation Network. The main purpose to produce this network was to calculate the Expert Cost as a parameter in ChemoTF. A citation network is a network of authors connected through the citations to/from their publications. This means that unlike the collaboration network, citation network is a directed graph. The direction of an edge represents the direction of the citation. In other words, given two experts x_a and x_b, the connection $x_a \rightarrow x_b$ suggests x_a cites x_b and vice versa. Apart from being directed, a citation network has similar attributes to the collaboration network. This is to say that this network is a directed, unweighted, and looped multigraph. It is looped to cover self-citations and it is a multigraph because the vertices are denoted to authors with multiple publications. Similar to the collaboration network, citation network accommodates 472,365 vertices, however it embodies 3,466,859 edges referring to the total number of citations.

The techniques employed to visualise the collaboration network are

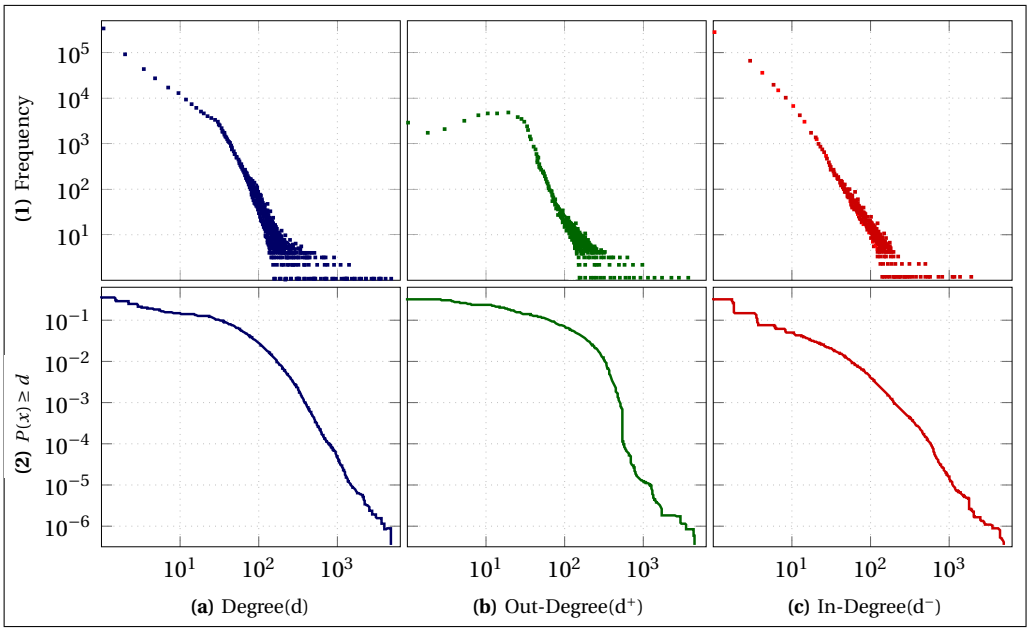


Figure 5.5: CompScholarCorp Citation Network Visualisation. Row 1 is Power Law Degree and Row 2 is Cumulative Degree Distributions.

also applicable to visualise the citation network. Since this network is directed, we visualise this network in 6 graphs on log-log scale including degree, out-degree and in-degree *Power Law Degree Distribution* depicted on row 1 of Figure 5.5, and degree, out-degree and in-degree *Cumulative Degree Distribution* demonstrated on row 2 of this figure. As observed from row 1, similar to collaboration network, the degree distribution of citation network is a power law. The power law exponent beginning with the minimum degree of $d_{min} = 51$ is estimated $\gamma = 3.28$. A negative correlation between degree and frequency is also identifiable which we calculated at the rate of -82.3%. In addition, vertices with lower degree than median average are abundant. This is because 88.1% of the authors have less than median average citations which is 8 for this network.

Comparing out-degree and in-degree distributions in graphs 1(b) and 1(c) of Figure 5.5, it is found that the boundaries of frequency for in-degree distribution are higher. This is because the number of citations gained by

some authors overwhelmingly exceeds the number of publications authors cite. In other words, while the median average ratio of in-degree to out-degree citations is $\frac{degree(d^-)}{degree(d^+)} = 4.45$, this ratio for highly influential authors who constitute 2.1% of the network is $\frac{degree(d^-)}{degree(d^+)} = 209.45$. It is also found that unlike in-degree distribution which stays a power-law with the goodness of fit of 98.2%, the out-degree distribution changes its behaviour as the frequency increases. This can be observed by comparing the steady growth of in-degree distribution shown in the Figure 5.5 graphs 1(c) and 2(c) to the top curve of out-degree distribution demonstrated in graphs 1(b) and 2(b). The out-degree distribution shows a gradual negative growth rate of $k = -1.21$ as the frequency raises. The steady growth of in-degree distribution suggests a homogeneous distribution of references used by authors which are dependant upon the number of publications. The curve on top of the out-degree function, however, suggests that the number of publications authors are cited for does not generally go beyond $\eta = 181 \pm 1$ and is likely to stay this way. As a rule of thumb, it can be stated that:

Statement 5.4.1. ***(Rule of Thumb for Citations)***

There is 87% *probability that a random selection from a citation network of any size will yield a publication with less than* $\eta = 181 \pm 1$ *citations.*

In this chapter I characterised the data in CompScholarCorp from the perspective of keywords and authors, and visualised the research communities derived from the corpus in the form of large collaboration and citation networks. I managed to achieve up to 98.2% goodness of fit for my model and revealed the characteristics such as trends, behaviour, scale of collaboration, scale of influence, and underlying relationships of research communities and other attributes featured in CompScholarCorp. The richness of this corpus makes it an asset with various research aims. However, the subjects covered in this chapter are limited to the use of CompScholarCorp as data source for ChemoTF experiment.

In the following chapter, I use probabilistic semantic analysis to derive latent topics of the publications and model existing relationships between authors. Moreover, I employ approximation techniques to compute the scale of contribution of each topic in relationships as well as the probability of belonging each publication to a topic. I will refer to *author* as *expert*, *publication* as *experience*, *keyword* as *skill*, and *topics* as *areas of expertise or AoE*. The rationale for using TF terminology is to provides better understanding of the problem an a clear picture of my solution.

"Mathematics is much more than a language for dealing with the physical world. It is a source of models and abstractions which will enable us to obtain new insights into the way in which nature operates."

Melvin Schwartz [114]

Chapter 6

Modelling and Analysis

The next stage of preparations for ChemoTF experiment involves modelling relationships between experts and constructing a network of experts which is inclusive of the scale of relationships in each area of expertise. In order to achieve this, I modelled relationships between experts using a specific instance of probabilistic generative topic models called *Latent Dirichlet Allocation* (LDA). I used the the concept of generated model to build discover latent areas of expertise and formed the relationship between experts in each area accordingly.

In order to calculate the scale of contribution of each area of expertise in relationships, I employed a machine learning inference technique called *Variational Inference*. Using this technique, I approximated the probability of belonging experiences to each area of expertise as well as the probability of appearance of skills in areas of expertise. Finally, in order to ensure that the generated model reflects the reality as expected, I evaluated the model using an approximation technique called *Perplexity*. I repeated the process of evaluation by adjusting the number of areas of expertise for each run and obtained a desirable model. Having a fit model in hand, I ran the analysis and generated the entities of ChemoTF. The next three sections of this chapter have been dedicated to elucidating the above mentioned topics.

6.1 Modelling relationships using LDA

The notion of relationships between experts lies in their collaborations. In an expert network where each expert is associated with various skills, the relationships are multinomial. This means that relationships encompass several aspects each providing a unique view of the previous experiences. In other words, these aspects which were defined as area of expertise in Chapter 3, make it possible to interpret relationship according to the circumstances and historical events.

Areas of expertise can be extracted using semantic analysis or other similar techniques, however the result of this type of analysis does not reflect the scale each area of expertise contributes to the relationships. In other words, the probability of each area of expertise appearing in a set of collaborations cannot be extracted from such analysis. This gap is the subject of a set of techniques called *Latent Dirichlet Allocation* (LDA) [22] which is a probabilistic form of Latent Semantic Indexing (pLSI) [62]. LDA is a generative probabilistic model for collections of discrete data (such as text corpora) in a semantic manner. It is considered a hierarchical *Bayesian* model based on a specific type of topic models, called *generative topic models* [66], in which each item of a collection is modelled as a finite mixture over an underlying set of topics. Documents are considered as mixtures of topics and each topic is defined as a multinomial probability distribution over words.

These models assumes that there is bag of documents with no structure (*i.e* meta-data, keywords or hierarchical information), and aim to discover the underlying structures by regenerating documents from observed words. To achieve this, LDA considers words inside documents as observations arisen from a generative probabilistic process and applies a sequence of probabilistic steps based on posterior inference. Then the inferred data is situated into the estimated model to see if a specific query about a new document can fit into the estimated topic structure.

There are several reasons why I chose LDA for the purpose of my analysis. The first reason relates to how LDA defines its components. LDA defines topics as multinomial space consisting of all the combination of words in the vocabulary list given their probabilities. In addition, documents in LDA are defined as random mixture of corpus wide topics. In other words, documents are probability distributions over topics and topics are probability distributions over words in the vocabulary list. This definition of documents, topics, and words in the vocabulary list in LDA can be interpreted as abstracts, research areas, and keywords for CompScholarCorp which correspond to the definition of experiences, areas of expertise, and skills in ChemoTF. This leads to the assumption that the word proportion of a topic in LDA can be considered as the probability of appearance of skills in each area of expertise. In addition topic proportion of a document in LDA can be considered as the probability of appearance of areas of expertise in each experience occurred during a contribution between a set of experts.

Another reason to use LDA for the purpose of my analysis was its simplicity and relatively good accuracy in modelling large datasets. One difficulty semantic analysis in modelling large datasets brings is the complexity of the topics, particularly in matrix factorisation based techniques such as *Latent Semantic Analysis* (LSA) [46]. These techniques assume that there is an entry for every topic in a document. Given that in reality documents generally contain a small number of topics, such an assumption is inapplicable to large datasets. LDA addresses this issue by adopting a generative approach and applying a *Dirichlet prior* so that the topics and their proportions can be learnt as the generative process continues to observe more documents. Given that CompScholarCorp contains over 1 million publications, LDA can be employed as an ideal technique to generate an accurate model.

The final reason for choosing LDA was its scalability and compatibility with fast inference techniques. Although LSA is considered faster in gen-

eral, the way LDA models large datasets allows the inference task to be performed in a parallel fashion. Methods such as Variational Inference are adjustable for this purpose. Considering that computing the inference is usually very costly, parallelisation can significantly decrease the computation time. In addition, the ability to evaluate the generated inference results of LDA by metrics such as *Perplexity* can potentially eliminate the necessity of using human-based evaluation techniques such as *word intrusion*. This feature can further accelerate the whole process.

As stated above, for the purpose of my analysis I considered experiences as documents, words which describe each experience as discrete observations of skills, and skills as the words in vocabulary lists. Blei et al [22] generate their vocabulary list by adding the most common words throughout the corpus into a list, computing the occurrence of every word in the list, while only considering words which exceed a particular threshold in terms of their frequency. Their list excludes stop words (*i.e* but, and, comma, full-stop, etc). Considering that skills are actually publication keywords which are the most common words in that publication reflecting the topic, the vocabulary list in the case of my analysis is the list of distinct skills throughout the corpus. The vocabulary size of my model in this research contains a $24,500$ unique keywords which reflect skills used in $1,044,454$ experiences. After I built the list, I used the generative probabilistic process of LDA as a machine learning technique to cluster discrete observations into areas of expertise. Then I used these clusters to label previously unlabelled experiences.

In order to describe the mechanism of LDA, it is essential to understand its main building block called *Dirichlet*. A Dirichlet is defined as a distribution over multinomial, which itself is a distribution over discrete outcomes [66]. A Dirichlet distribution is usually parameterised by a vector, often denoted by α, and described by a multivariate continuous probability distribution in the form of a normalising constant, often denoted by Γ. The parameter α controls the scale and the centrality of a Dirichlet.

Equation 6.1 is the formal definition of a Dirichlet distribution.

$$Dir(\alpha_1,...,\alpha_T) = \frac{\Gamma(\sum_j \alpha_j)}{\prod_j \Gamma(\alpha_j)} \prod_{j=1}^{T} p_j^{\alpha_j - 1} \tag{6.1}$$

I have illustrated the mechanism of LDA using a plate notation in Figure 6.1. In this graphical model, random variables are represented by nodes and the possible dependencies between them are denoted by edges. The reason for shading the skill node is that skills in each experience are the only components of the model which are directly observed whereas other components are hidden and cannot be directly observed. In addition, the structure of the graph defines the pattern of conditional dependence between the ensembles of random variables that the nodes represent. The indices in the boxed indicate the number of replication of each random variable.

To cast this model into a generative probabilistic process, considering the definition of the Dirichlet explained earlier, I adjusted the original LDA algorithm to reflect the definitions of variables in ChemoTF and applied it into CompScholarCorp. This adjustment has been portrayed in Algorithm 3 where the modelling has been performed for each observed skill of each experience.

Apart from parameters α and λ of Dirichlet prior and skills $s_{e,n}$, the rest of the components ($z_{e,n}$, θ_e, β_k) are hidden variables and often called latent variables. These are the variables that need to be inferred in step 2 of the given algorithm. Given that computing the posterior for these variables is complex, approximation techniques are often employed instead. Techniques such as *Mean-Field Variational* [22], *Expectation Propagation* [101], and *Gibbs Sampling* [66] are probably the most significant methods in this area. One common thing among all these techniques is that they all try to approximate latent variables through a process in which parameters are learnt and adjusted to best fit in the model.

For the purpose of this analysis and in order to accomplish the anal-

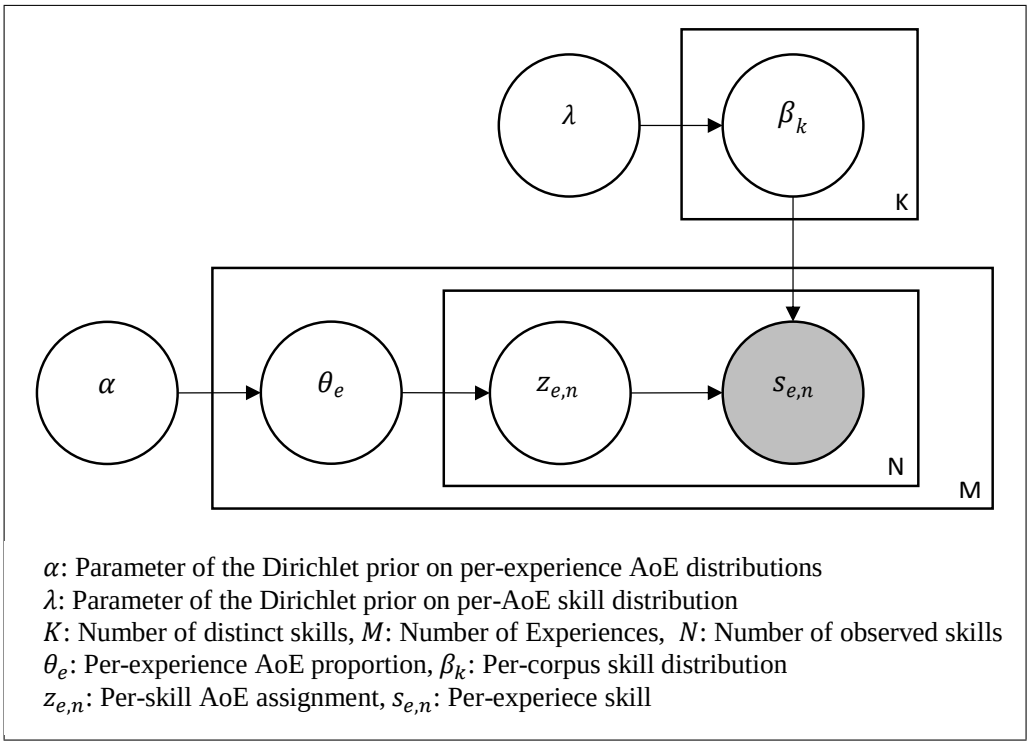


Figure 6.1: LDA Plate Notation adjusted for ChemoTF

ysis task, I adopted Variational Inference [22]. The next section explains why this particular technique was adopted and elaborates my analysis.

6.2 Analysis using Variational Inference

Variational Inference is a technique for approximating intractable integrals which arise in Bayesian inference and machine learning [75]. This technique is typically used in complex statistical models consisting of observed variables as well as unknown parameters and latent variables.

The aim of Variational Inference is to provide an analytical approximation to the posterior probability of the unobserved variables, in order to perform statistical inference over complex distributions that are difficult to directly evaluate. An alternative to Variational Inference to perform

Algorithm 3: Adjusted LDA Algorithm For ChemoTF

1 **foreach** *area of expertise* ***in*** *corpus* **do**

/* draw a multinomial distribution β_k from a Dirichlet distribution with parameter λ */

$\beta_k \sim Dir(\lambda) \Rightarrow p(\beta_k|\lambda) = \beta_{k,\lambda}$

2 **foreach** *experience* ***in*** *corpus* **do**

2a /* draw a multinomial distribution θ_e from a Dirichlet distribution with parameter α */

$$\theta_e \sim Dir(\alpha) \Rightarrow p(\theta_e|\alpha) = \frac{\Gamma(\sum_i \alpha_i)}{\prod_i \Gamma(\alpha_i)} \prod_k \theta_{e,k}^{\alpha_k - 1}$$

foreach *skill* ***in*** *vocabulary list* **do**

2b /* select a hidden area of expertise z_n from the multinomial distribution parameterized by θ */

$z_{e,n} \sim Mult(\theta_e) \Rightarrow p(z_{e,n}|\theta_e) = \theta_{e,z_{e,n}}$

// select observed skills $s_{e,n}$ from the distribution βz_n

$s_{e,n} \sim Mult(\beta_{z_{e,n}}) \Rightarrow p(s_{e,n}|\beta_{z_{e,n}}) = \beta_{z_{e,n},s_{e,n}}$

3 **return**$(z_{e,n}, s_{e,n})$

such a task is Markov Chain Monte Carlo (MCMC) [8] oriented methods such as Gibbs sampling [61]. The difference basically lies in the form of approximation each provides. Gibbs Sampling provides a numerical approximation to the exact posterior using a set of samples, whereas Variational Inference provides a locally-optimal, exact analytical solution to an approximation of the posterior [20]. This difference between the two method along with the two main reasons bellow constitute the rationale behind my preference of Variational Inference over Gibbs Sampling.

The first reason for choosing Variational Inference is that it is deterministic. This means that, unlike MCMC oriented inference techniques, there is no randomness involved in the nature of Variational Inference and the outputs are always the same for identical data in multiple runs.

This is an advantage as I ran the analysis and evaluated the results multiple times. The evaluation was particularly useful to judge how well the constructed areas of expertise represented the experiences and the relationships. Therefore, I was able to make a decision for the number of AoEs in a way that best fit the model. In addition, the requirement of this research demanded devising a method that could be compared with other approaches, hence Variational Inference outweighed other inference techniques such as Gibbs Sampling.

The second advantage of Variational Inference over other techniques is that it is easier and faster to gauge convergence. In other words, it is clear when it reaches the answer. The reason is that Variational Inference only requires a small number of iterations to accomplish the inference task. In addition, Variational Inference can be easily devised in a parallel manner where steps in its algorithm are processed diversely and independently regardless of outcome of each step. This was an important advantage unique to Variational Inference which significantly reduced the process time particularly for my relatively large dataset. Although other inference methods such as MCMC-oriented algorithms are considered faster per step, the overall computation time of Variational Inference for large-scale datasets has been said to be lower in general [75].

The purpose of using Variational Inference was to adjust its inference function and estimate the latent variables. Variational Inference takes advantage of an objective function $\mathscr{L}$ with four parameters including γ,φ,α, and β and is defined as demonstrated in Equation 6.2. Appendix A elucidates this function and explains its origin.

$$\begin{aligned}\mathscr{L}(\gamma,\varphi;\alpha,\beta) = \mathbb{E}_q[\log p(\theta|\alpha)] + \mathbb{E}_q[\log p(z|\theta)] + \mathbb{E}_q[\log p(s|z,\beta)] \\ - \mathbb{E}_q[\log q(\theta)] - \mathbb{E}_q[\log q(z)]\end{aligned} \tag{6.2}$$

Assuming that Variational distribution in the case of my thesis is mean-field and all the AoE assignments are derived from a multinomial distribution φ, this objective function reflects the inference function I used to perform the analysis. Following the work in [22], I configured the original

Variational algorithm to solve the objective function given in Equation 6.2. Algorithm 4 elaborate my adjusted version of Variational Inference for ChemoTF. In this algorithm, The parameters are estimated and the results of the convergence are examined in each round. The process is repeated for each skill in the each experience with respect of the AoE and stops when the convergence is reached. The calculation of expectations and update functions in the given algorithm requires an extensive mathematical background and elaborations. For the purpose of integrity of this thesis, the discussion and formulas involved in the calculations have been moved to Appendix B.

Using the Algorithm 3, I first fit the corpus data into a generative topic model that I already constructed and then concluded the analysis by performing the steps described in Algorithm 4. In order to evaluate the fitness of the model and accuracy of the analysis, I obtained a method called "perplexity" which has been expounded in [22]. The last section of this chapter delineates the undertaken evaluation and demonstrates the final results in details.

6.3 Evaluation and Results

The final task in modelling is to ensure that the model is reliable and reflects the reality as expected. To achieve this in LDA, two typical methods have been developed to determine the goodness of fit of the model. The first method is known as "evaluation of clustering result" [22]. This technique evaluates the model by measuring its performance on multiple secondary tasks. The quality of the model is specified by various metrics such as the semantic relationship between each latent variable, the similarity of inter-class documents, etc. Although this method encompasses comprehensive evaluation criteria, it requires multiple datasets or dividing the dataset into various sub-datasets which is out of the scope of this thesis.

Algorithm 4: Adjusted Variational Inference Algorithm For ChemoTF

initialize (**random**(γ , φ))

foreach *iteration i and j* **do**

  foreach *experience **in** corpus* **do**

    /* update Dirichlet distribution γ_i and multinomial $\varphi_{n,i}$ by solving the first partial derivative of the objective function for γ_i and $\varphi_{n,i}$ */

    $\frac{\delta \mathscr{L}}{\delta \gamma_i} \overset{udpate}{\Longrightarrow} \gamma_i = \alpha_i + \sum_n \varphi_{n,i}$

    $\frac{\delta \mathscr{L}}{\delta \varphi_{n,i}} \overset{udpate}{\Longrightarrow} \varphi_{n,i} \propto \beta_{i,v} \mathbb{E}(\Psi(\gamma_i) - \Psi(\sum_j \gamma_j))$

    // Ψ is the digamma function, first derivative of $\log\Gamma$

  foreach *area of expertise **in** corpus* **do**

    /* update AoE distribution β by solving the first partial derivative of the objective function for $\beta_{i,j}$ */

    $\frac{\delta \mathscr{L}}{\delta \beta_{i,j}} \overset{udpate}{\Longrightarrow} \beta_{i,j} \propto \sum_e \sum_n \varphi_{e,n,i} s^j_{e,n}$

  if ***converged**(**compute**($\mathscr{L}(\gamma, \varphi; \alpha, \beta)$))* **then break**

return($\mathbb{E}_\gamma, \mathbb{E}_\varphi$)

The second method is commonly referred to as "likelihood of held-out data" [128]. It calculates the probability of held-out data that are not used for training. This probability is proven intractable hence various approximation techniques have been developed to estimate it. Wallach et al [128] summarise several estimation methods such as *importance sampling method, harmonic mean method, annealed importance sampling,* and *perplexity.* For the purpose of this thesis I adopted perplexity as described in [22]. The decision was made to avoid the mathematical complexity involved in other estimation techniques.

Perplexity is defined as the reciprocal geometric mean of the token likelihoods in the test corpus [22]. Equation 6.3 describes this definition for my testbed, where M is the number of experiences, s_e represents the skills involved in each experience e, and N_e is denoted to the number of skills in each experience. Lower values of perplexity indicate lower misrepresentation of the skills in experiences by the generated areas of expertise. Hence, the lower the value of perplexity gets, the more reflective of data the model becomes.

$$Perplexity = \exp\left\{-\frac{\sum_{e=1}^{M} \log p(s_e)}{\sum_{e=1}^{M} N_e}\right\} \tag{6.3}$$

As it can be concluded from the Equation 6.3, the number of areas of expertise plays an important role in perplexity value. The right number of hidden areas of expertise results in lower perplexity value thus a desirable model and a reliable analysis. For this reason, I followed the method explained in [22] and used perplexity after each round of test to verify the number of areas of expertise. I held out 10% of the data for testing purposes and trained the models on the remaining 90%. I repeated the process of evaluation until the perplexity returned its lowest value. Figure 6.2 displays the evaluation results.

As observed from Figure 6.2, the evaluation was started by 10 areas of expertise. Increasing this number on each test had a positive effect on perplexity value until the the perplexity gets to its lowest value which is 3490 in my case. The number of areas of expertise when the perplexity returns the lowest value for the first time is 140. Further increase in the number of areas of expertise had no effect on perplexity value as it remained low for the duration of the test with 200 areas of expertise respectively. This indicates that any number of areas of expertise between 140 to 200 would make the model fit for my experiment. Therefore, I specified 160 areas of expertise for my experiment.

Having a fit model and the number of areas of expertise in hand, I performed the analysis on the whole corpus. I began by eliminating the

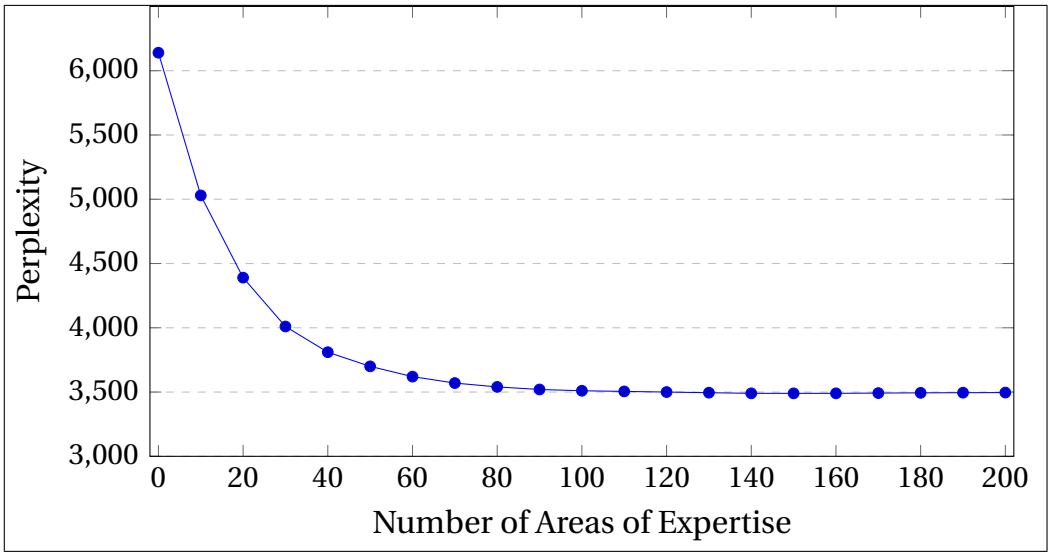


Figure 6.2: Perplexity results on CompScholarCorp

stop words in the corpus. Then I used Algorithm 3 to find the Dirichlet and conditional multinomial parameters for my LDA model. Then I concluded the analysis by using Algorithm 4 to compute the Variational posterior Dirichlet parameters for all experiences and Variational posterior multinomial parameters for each observed skills.

Figure 6.3 illustrates applying my model and performing analysis on the example I discussed in Section 4.3.1 of the previous chapter. In this figure, the top skills from some of the resulting multinomial distributions are illustrated at the top. These skills capture some of the underlying areas of expertise in the corpus, thus I have named them according to these areas of expertise. Due to the space limitation, only 9 skills with the highest probability of appearance have been demonstrated for each area of expertise whereas in reality, each area of expertise contain all the skills with their probability of appearance. At the bottom of the figure, the experience of my example has been illustrated. Each observed skill correspond to one or more areas of expertise which are highlighted accordingly. For simplicity, I have only demonstrated 4 areas of expertise in which the ob-

served words have higher probability of appearance.

To this end, I first identified the areas of expertise of each experience. Then I computed the probability by which each experience belongs to each area of expertise. I then formed the relationships according to this probability value. In addition, following the definition given for relationships in Chapter 3, I formed skill-oriented relationships by embedding areas of expertise and their probability values into the mutual experiences between experts. Therefore, the relationships mirror various unique aspects according to the skills chosen from each expert.

Finally, in order to avoid complexity and echo a real-world simulation of experts, I specified the most frequently used skills for each experts and represented experts with their top skill-set. Given that experts represent themselves with a small number of their top skills in real life, my simulation of experts with limited top skills reflect a better representation of reality. Therefore, I condensed the long list of skills for each expert into a skill-sets with the maximum number of 6 skills of the highest frequency of appearance. This filter was only applied for skill-set of experts meaning that the number of skills in areas of expertise remained untouched as they will be required for calculating metrics such as Chemistry Level, Communication Cost, and Expertise Level.

To conclude this chapter, I modelled the relationships using LDA and generated 160 areas of expertise to cover 2400 unique skills in the vocabulary list. I quantified the scale of relationships and calculated the probability of each experience belonging to areas of expertise as well as the probability of skills appearing in area. The next chapter describes how I designed and implemented a software to experiment ChemoTF as well as the decisions I made during this process.

"SOCIAL MEDIA"	"MOBILITY"	"WIRELESS NETWORK"	"SECURITY"
SOCIAL NETWORK	MOBILE	INTERNET	PROTOCOL
SHARING	PHONE	AD HOC	SECURITY
GRAPH	ANTENNA	SENSOR	CRYPTOGRAPHY
USER	MIDDLEWARE	IP	CIPHER
SEARCH ENGINE	NAVIGATION	COGNITIVE	IDENTITY
COMMUNITY	BANDWIDTH	TOLERANCE	PRIVACY
SOCIETY	SIGNAL	MESH NETWORK	PUBLIC KEY
FRIEND	REAL TIME	ROUTING	ENCRYPTION
SEMANTIC	MOBILE ROBOT	ACCESS POINT	DATA ACCESS

Mobile social networks (MSNs) are specific types of social media which consolidate the ability of omnipresent connection for mobile users/devices to share user-centric data objects among interested users. Taking advantage of the characteristics of both social networks and opportunistic networks (OppNets), MSNs are capable of providing an efficient and effective mobile environment for users to access, share, and distribute data. However, the lack of a protective infrastructure in these networks has turned them into convenient targets for various perils. This is the main impulse why MSNs carry disparate and intricate safety concerns and embrace divergent safety challenging problems. In this paper, we aim to provide a clear categorization on safety challenges and a deep exploration over some recent solutions in MSNs. This work narrows the safety challenges and solution techniques down from OppNets and delay-tolerant networks to MSNs with the hope of covering all the work proposed around security, privacy, and trust in MSNs.

Figure 6.3: An example experience as appears in CompScholarCorp. Each colour codes a different factor from which the skill is generated.

> "Programming is the art of organizing complexity, of mastering multitude and avoiding its bastard chaos as effectively as possible."
>
> *Edsger W. Dijkstra* [42]

Chapter 7

Implementation

Having prepared and modelled all the required components for the experiment, the last step to experiment ChemoTF is implementation. The implementation in the case of this thesis involves developing a software in order to experiment ChemoTF along with three well-known TF algorithms. This software produces reliable and reputable results that can make it possible to conduct a comparative study in an extendable, repeatable, and efficient manner. This chapter discusses the steps taken to develop such software. These steps include requirement analysis, construction of a relational database, implementation of core functionalities, and expanding the implementation to cover all requirements.

In the first section, a requirement analysis is conducted to identify the core objective and mechanism of each algorithm. In addition, the requirements regarding the implementation techniques and technologies are identified and decisions are made accordingly. In the second section, a database was designed and built to facilitate the process of providing necessary data from CompScholarCorp for the algorithm during the experiment. In the third section a test harness was built and the identified core objectives were implemented and verified. In the final section, the implementation was expanded to cover all the requirements, including data and functionalities, and the final software was developed.

7.1 Requirements Analysis

The software requirements are a series of descriptions regarding objectives and functionalities of required software. The process of gathering, analysing and documenting these requirements is known as *requirement analysis* (often called requirement engineering) [65]. Requirement analysis has traditionally been one of the first steps of any software development project. The aim of such analysis is to clarify the expectations of a software product by an expository document which includes comprehensive list of declarations of what the project is supposed to achieve [83].

I began the process of requirement analysis by summarising the core objectives. Part of these objectives includes the requirements of my algorithm as the main contribution of this thesis. Given that I had already gathered the required data and compiled CompScholarCorp, designed and built a database, and modelled the social network, the scope of my implementation was clear and contained all the requirements, metrics and functionalities discussed in previous chapter. The aim of implementation with regard to ChemoTF was to develop a reliable, reusable, and scalable software to experiment ChemoTF in fast and accurate fashion.

In order to demonstrate that my algorithm is an improvement over the existing TF algorithms, I adopted a comparative approach in which the experimental results of ChemoTF is evaluated against some of the well-known TF algorithms. This comparative approach opened up a new set of requirements. It was found necessary to implement these algorithms as well as ChemoTF because their corresponding implementation codes have not been made available by their authors. Although the comparative study conducted by [130] implements these algorithms in TF and promises a repeatable experiment, I could neither replicate the experiment nor obtain the code from the authors. A significant part of the implementation is missing from the code made available by the authors thus their implementation could not be utilised.

For the purpose of this thesis, I chose three well-know algorithms to implement and compare their results against ChemoTF. These algorithms include EnSteiner [86], MinSD [76], and RarestFirst [86]. The main reason for selecting these algorithms is their considerable influence among research conducted in the area of TF. These algorithms have been used in [7, 51, 57, 76, 78, 89–91, 119, 129, 130, 137] for the purpose of comparative analysis. All the three algorithms are from the first category of TF algorithms discussed in Chapter 2 aiming to find teams with minimum Communication Cost. Given that the three algorithms mentioned above define Communication Cost differently, the main requirement in my implementation was to ensure that the Communication Cost function generated valid and accurate results as expected.

The rest of this section has been dedicated to describing the comparative algorithm namely EnSteiner, MinSD, and RarestFirst. For each algorithm, a short background has been given, the mechanism of forming the final team has been expounded, the definition of Communication Cost has been elucidated, and the time complexity has been described. In addition, the decisions made with regards to the techniques, and technology to achieve a high-quality implementation has been described.

7.1.1 Comparative Algorithm 1: EnSteiner

The first algorithm I chose to conduct experimental comparisons against was *Enhanced Steiner*(EnSteiner) [86]. It is well-known algorithm and has been inspirational for much research in the area of TF in social networks. The EnSteiner algorithm is motivated by the obvious similarity between the *Minimum Spanning Tree* problem and the *Group Steiner Tree* problem [87]. This algorithm is best known for finding cohesive sets of experts thanks to its deviation mechanism conducted by *Steiner-Tree*. In addition, the greedy nature of this algorithm makes it possible to form teams with small size even although such an achievement comes with a draw-

back of extensively high processing time. This drawback is noticeable in large networks where the iterations of Spanning Steiner-Tree are generally high in number.

Algorithm 5 presents the mechanism employed in EnSteiner algorithm. Given a task T, the process begins in step 1 by generating an enhanced graph H from graph G according to the task. This graph encompasses only experts that posses required skills. Then in step 2, an expert from this graph is randomly chosen. In step 3 to 6, the algorithm adds the undirected edges between experts and their corresponding skill vertex based on the minimum Communication Cost of the artificially added edge. In step 7, these edges along with their nodes at both ends are removed and the solution X_H is obtained as the final team.

The Communication Cost in this algorithm is defined as the sum of pairwise Communication Costs of graph G. In other words, Communication Cost for EnSteiner algorithm is the weight cost of the minimum spanning tree for graph G which is calculated as demonestrated in Equation 7.1 where x_i and x_{s_j} are the vertices in H and correspond to x_n and x_m in social graph G.

$$ComCost(x_i, x_{s_j}) = \sum d(x_m, x_n) \tag{7.1}$$

The overall running time of EnSteiner is $O(k \times |E|)$ where k is the number of iterations in a spanning tree and $|E|$ is the number of edges in the social network. There have been some efforts to reduce this time complexity by employing other approximation techniques. Authors in [58] achieve an approximation ratio of $O(\log^3 n log k)$ where n refers to the number of nodes. In the case of G being a tree, this complexity can be reduced to $O(\log^2 n)$. However, given that the nature of social networks demands it to be a graph, this problem is not approximable within $\Omega(\log^{2-\epsilon})$ unless *NP* admits quasi-polynomial-time *Las Vegas* algorithm. [67].

In the following section, I demonstrate the mechanism of MinSD as the second comparative algorithm.

Algorithm 5: EnSteiner for TF problem regenerated from [86]

Input: Social Network $G(X,E)$, A pool of Expert X each with multiple skills $s \in S$, and a Task T

Output: An expert team to cover T

(1) $H(X_H, E_H) \leftarrow EnhancedGraph(G,T)$
(2) $X_R \leftarrow Random(\{x | x \in X_T\})$
(3) **while** $(X_R \setminus X_H) \neq \varnothing$ **do**
(4) $\quad ComCost(x,s) \leftarrow Distance(X_H, s)$
(5) $\quad selectedExpert \leftarrow arg\ min_{s \in T \setminus X_H} ComCost(x)$
(6) $\quad$ **if** $Path(selectedExpert, X_H) \neq \varnothing$ **then**
(7) $\quad\quad X_H \leftarrow X_H \cup \{Path(selectedExpert, X_H\}$
(8) $team \leftarrow X_H \setminus T$
(9) **return** $team$

7.1.2 Comparative Algorithm 2: MinSD

The second algorithm I chose to conduct experimental comparisons against was *Minimum Sum of Distance*(MinSD) [76]. This algorithm is an improvement over *Minimum Spanning Tree* (MST) algorithm proposed in [86]. It addresses the unstable nature of MST in a dynamic social networks and proposes a new function to calculate Communication Cost based on minimum distance to the spanning tree. The mechanism of MinSD has been explicated in Algorithm 6.

As demonstrate in this algorithm, the first step initialises the desired Communication Cost for the team that algorithm aims to form (denoted by $leastSumComCost$) to $+\infty$. This step guarantees an answer for the algorithm. The algorithm then goes into a nested loop in step 2, and 3. For each skill in the task T and for each expert x possessing the chosen skill, the sum of Communication Cost is initialised to zero in step 4 and the chosen expert is added into candidates pool along with his or her skill s_i in step 5. Then in the steps 6 to 10, for each one of the other required

skills (denoted by s_j), the algorithm first calculates the Communication Cost of each expert x given s_j, and adds the closest neighbour to the candidates pool. In addition, $sumComCost$ is updated by adding the Communication Cost of the candidate and the selected expert. Finally in steps 11 to 13, if Communication Cost of the discovered team is the lowest ever found, the team is replaced by the candidates.

MinSD defines Communication Cost based on distance between experts and derives it from $Distance$ function. This function returns the shortest path between an expert x and the set of experts $X(s_j)$ consisting of all the experts who posses the skill s_j. Equation 7.2 formulates this Communication Cost. Taking advantage of the $Neighbour$ function which returns the nearest neighbour to an expert x in $X(s_j)$, the algorithm ensures that the candidates are selected based on their shortest path to their nearest neighbour.

$$ComCost(x_i, x_j) = min\ d(x_i, x_j) \tag{7.2}$$

The time complexity of MinSD is $O(|T|^2 \times X^2_{max})$ where $|T|$ is the number of required skills (cardinality of the task) and $|X_{max}|$ is the maximum number of generated sub-graph of experts possessing a particular skill $|X_{max}| = max|X(s)|$. The value $|X_{max}|$ is the complexity of $Distance$ and $Neighbour$ functions pre-computed for all pairs and stored in a hash table. The authors indicate that in the case of the hash table becoming very large, other database indexing techniques suggested in [63] can be employed. In the worst case scenario where the number of sub-graphs is equal to the n number of experts $|X_max| = O(n)$, the running time increases to $O(|T|^2 \times n^2)$. However in real data sets $|X_max|$ is much smaller than n [86].

The next section is devoted to describing RarestFirst as the last algorithm I implemented to evaluate my algorithm against.

Algorithm 6: MinSD for TF problem regenerated from [76]

Input: Social Network $G(X,E)$, A pool of Expert X each with multiple skills $s \in S$, and a Task T

Output: An expert team to cover T

(1) $leastSumComCost \leftarrow +\infty$
(2) **for** $i \leftarrow 1$ **to** $|T|$ **do**
(3) **foreach** *expert* $x \in X(s_i)$ **do**
(4) $SumComCost \leftarrow 0$
(5) $candidates \leftarrow \{\langle s_i, x\rangle\}$
(6) **for** $j \leftarrow 1$ **to** $|T| : j \neq i$ **do**
(7) $ComCost(x, s_j) \leftarrow Distance(x, X(s_j))$
(8) $selectedExpert \leftarrow Neighbour(x, X(s_j))$
(9) $sumComCost \leftarrow sumComCost + ComCost(x, s_j)$
(10) $candidates.Add(\langle s_j, selectedExpert\rangle)$
(11) **if** $sumComCost < leastSumComCost$ **then**
(12) $leasSumComCost \leftarrow sumComCost$
(13) $team \leftarrow candidates$
(14) **return** $team$

7.1.3 Comparative Algorithm 3: RarestFirst

The last algorithm I have chosen to conduct comparison against is called RarestFirst [86]. It is an instance of *Multichoice Algorithms* [9] and has been influential among the researchers in the field of TF. One reason for such influence comes from its simplicity which facilitates the capability of running this algorithm in a distributed fashion. Authors in [130] accomplished this by using a cloud based platform. Another reason for this influence is the fast nature of this algorithm. Although the quality of the teams formed by RarestFirst are questionable, the ability to obtain rapid answers has been popular among researchers to some extends.

Algorithm 7 demonstrates the mechanism employed in the original RarestFirst algorithm. For each skill s_a, the algorithm first generates $X(s_a)$ which is a set of experts possessing the skill s_a. Step 1 and 2 of the given algorithm depict this action. Then in step 3, given a task T, the algorithm chooses the rarest skill possessed among all the experts. In other words, the first skill s_{rare} this algorithm chooses has the lowest cardinality of $X(s)$. This is most likely why the algorithm has been called RarestFirst by the authors. Then the algorithm in step 4 to 7 computes the Communication Cost for all experts in $X(s_{rare})$ and chooses the expert with the minimum diameter. For other required skills s_b, the algorithm chooses experts with the lowest communication compared to $X(s_b)$ as well as the experts on the shortest path. The communication function employed in this algorithm is defined as the longest shortest path between any experts in team and can be calculated as shown in Equation 7.3.

$$ComCost(x_i, x_j) = max_{x_i, x_j} min\ d(x_i, x_j) \tag{7.3}$$

The running time of RarestFirst is $O(|X(s_{rare}) \times N|)$ while the worst-case can get to up to $O(N^2)$. However, Wang *et al.* [130] argue that with a slight change the worst-case running time can be reduced to $O(N)$. They suggest that instead of iterating through $X(s_{rare})$, every expert with the rarest skill can be assigned to one node in the cloud for which the Communication Cost can be computed separately. This can potentially decrease the time complexity unless the number of experts with rare skills is much larger than the nodes in the cloud.

I presented all three algorithms I chose to implement and described the mechanism involved in each algorithm. In the following section, I go through the important decisions I made before initiating the implementation. These decisions include choosing appropriate techniques, technologies and tools to meet the requirements and adjust the implementation accordingly.

Algorithm 7: RarestFirst for TF problem regenerated from [86]

Input: Social Network $G(X, E)$, A pool of Expert X each with multiple skills $s \in S$, and a Task T

Output: An expert team to cover T

(1) **foreach** *skill* $s_a \in T$ **do**
(2) $\quad X(s_a) = \{x_a | s \in X\}$
(3) $s_{rare} \leftarrow arg\ min_{s_a \in T} | X(s_a)$
(4) **foreach** *expert* $x \in X(s_{rare})$ **do**
(5) $\quad$ **foreach** *skill* $s_b \in T - \{s_{rare}\}$ **do**
(6) $\quad\quad ComCost(x, s_a) \leftarrow Distance(x, X(s_b))$
(7) $\quad ComCost(x, s_b) \leftarrow max_{s_b} ComCost(x, s_b)$
(8) $selectedExpert \leftarrow arg\ min\ ComCost(x)$
(9) $team = selectedExpert \cup \{Path(selectedExpert, X(s) | s \in T\}$
(10) **return** $team$

7.1.4 Techniques and Technology

Studying the mechanism of ChemoTF along with the comparative algorithms including EnSteiner, MinSD, and RarestFirst, it became evident that the algorithms operate very differently. This dissimilarity demands a separate set of validations for each algorithm. Furthermore, it was found that each algorithm defines Communication Cost in its own way. Therefore the implementation is required to a offer a separate Communication Cost function for each algorithm. In addition, conducting a fair and accurate experiment demands a data structure which can be equally accessible for all the algorithms. This necessitates an integrated platform where all the algorithms can run independently with the same input.

The above mentioned findings encompass the core objectives of my implementation. In order to achieve these objectives, I adopted Test-Driven Development (TDD) approach. The decision was made for two key reasons. The first reason was that TDD provides the capability of ac-

curate implementation in small steps. This is particularly useful when the projects has the potential to be decoupled into smaller units. Given that the test cases were transparent in my case, TDD could serve as a good software development techniques to be employed. The second reason was that the fast approach TDD offers for implementing key requirements. Given the scale of work in my thesis and the short deadline for the implementation, TDD was an ideal technique to rapidly achieve the desirable and accurate results.

Adjusting test cases to mirror the key objectives, TDD ensures that the implementation satisfies the core functionalities. These test cases can also be adjusted to serve as validation units. In this case, the expected results are turned into test cases and functions are implemented to pass these cases. Equipped with a test harness where all the algorithms can be developed in interactive and evaulable modules, TDD integrates the implementation of key functions with the validation which accelerates the implementation process for all three comparative algorithms.

In the case of ChemoTF functions including Chemistry Level, Expertise Level, and Expert Cost, the implementation must ensure that these functions can also be utilised by comparative algorithms. This is because the result of these functions are used as metrics to assess the quality of final teams. In order to achieve this, I adopted Object-Oriented Programming (OOP) approach. The main reason for this decision is that OOP is one of the best techniques to implement functions with similar objectives but different mechanism. Thanks to the concept of *inheritance*, a base class is defined to cover the similar attributes and each algorithm inherits them according to its mechanism. For dissimilar attributes, each algorithm overrides the attributes that have been inherited from the base class. Furthermore, the required data is defined as a public class which is accessible for all the algorithms during the run time.

In addition to TDD and OOP which I chose as my implementation techniques, I also chose*Relational Databases* as an efficient way to pro-

vide data for the implemented algorithms. Although the XML-oriented corpus described in Chapter 4 can fulfil this requirement, Relational Databases provide a more logical form of data structure which is both easier and quicker to interact with thanks to the tools, often referred to as *Relational Database Management Systems* (RDBMSs), offered for them [41]. Therefore I designed a schema to model CompScholarCorp along with the results of analysis described Chapter 6.

Another important decision made during the process of requirement analysis was with regards to the technology through which I implement the algorithms and database. For the database, I chose Microsoft SQL Server 2014 [1] as the RDBMS. The decision was made to avoid any compatibility issues with the ChemoTF development which at that stage was decided to be implemented in Microsoft .Net [2]. The language I used for the implementation was Microsoft C#.Net version 4.5. Since C# is an OOP language and can be easily used for TDD using *Test Explorer* offered in Microsoft Visual Studio (VS)[3], this language was ideal for my implementation. In addition, the built-in functionalities of VS for visualising the code through *Class Diagrams* facilitated software development process.

To this end, I adopted Relational Database to feed the algorithms, TDD to implement the core functions and OOP to expand the implementation to its final stage. I also chose Microsoft SQL Server and C#.NET to achieve such implementation. In the following section, I describe how I designed a database to make practical use of CompScholarCorp. Then I explain how I turned the core objectives into test cases using Test-Driven Development (TDD). In addition I expound how I set up an integrated platform to run each algorithm independently with the same input. Finally I demonstrate how I implemented such functionalities to pass the test cases and how I validated them against the previously obtained results.

[1] www.microsoft.com/en-us/sql-server/sql-server-2016
[2] www.microsoft.com/net
[3] www.visualstudio.com/

7.2 Relational Database

Although a well-designed XML-oriented dataset can be utilised in implementation, I used the Relational Database [34] for the purpose this implementation. The reason for my preference was that a Relational Database provides a data structure which is both easier and quicker to interact with. It is a collection of data structured in terms of schema, tables, queries, reports, views, and other objects and encompasses a model which reflects relationships between various entities of the structure.

I started the process by listing required fields, specifying the definition of each field, and omitting any unnecessary data. Then I matched the definition of each field with the definition given in the TF terminology and renamed the field accordingly. In addition, I broke the data into logical pieces and derived the relationships between each field. In doing so, I organised the data into columns and tables and drew the relationships between tables. In order to reduce data redundancy and improve data integrity, I used the normalisation technique suggested in [35] and produced a database in 3rd normal form (3NF). Each table has a *primary key* and is connected to other tables via *foreign keys*. The primary keys are marked with the word "ID" as part of their names, and foreign keys are named to end with "Ref" respectively. The names of the other fields were also carefully selected and reviewed so that they become self-descriptive.

Figure 7.1 depicts the schema of this database describing its different entities and the relationships between them. As observed from this figure, some of the fields of this database do not correspond to any field in CompScholarCorp (*i.e* TableAoE, TableSkillAoe, TableExperienceAoE). This is because the database includes post-analysis information done in Chapter 6 as well as the corpus data.

To this end, I built a database to be utilised throughout the experiment as a common gateway for all the algorithms. The rest of this chapter has been dedicated to describing the implementation process.

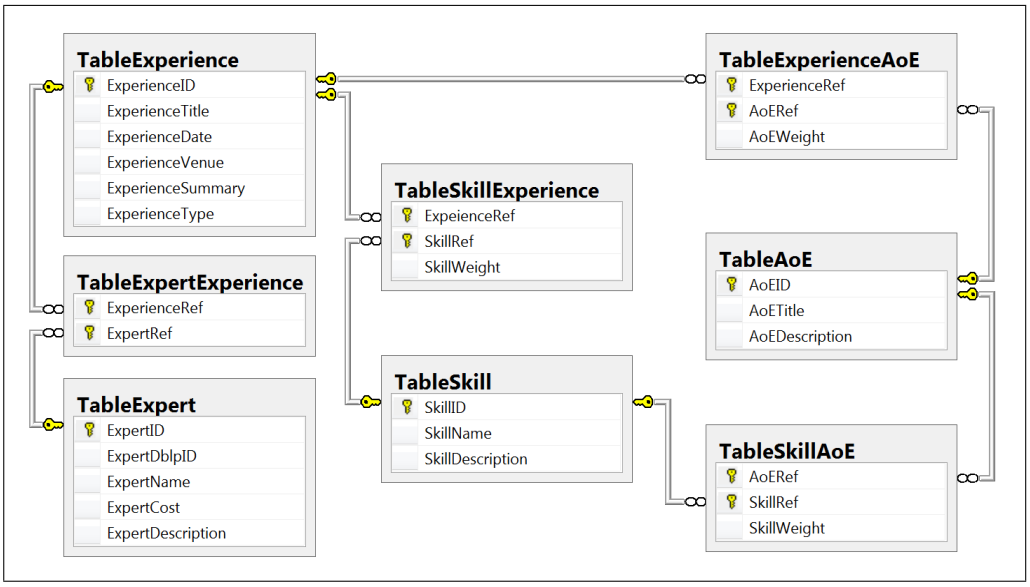


Figure 7.1: The Schema of ChemoTF Relational Database

7.3 Ticking Objectives: A TDD Approach

Test-Driven Development (TDD) is a software development process that uses test cases as requirements for which software are developed. [17]. The aim of TDD is to prioritise the design of the core requirements over the implementation of software features. From this perspective, TDD can be considered an *Agile* software development technique [98]. TDD process relies on the repetition of short development cycles. In each cycle the requirements are turned into specific test cases often called *unit tests*. The goal in each cycle is to implement just enough functionality to improve the software by passing the new tests only. The new cycle only begins if all the tests are passed. Figure 7.2 illustrates a single cycle in TDD.

As depicted in this figure, the process begins by writing a functional test case. It is important to run the test prior to any implementation and ensure the test fails in order to guarantee that the test case is correct. Then, the process of "writing minimal code" begins. In this step, the functional test case is broken into one or more smaller unit tests and a mini-

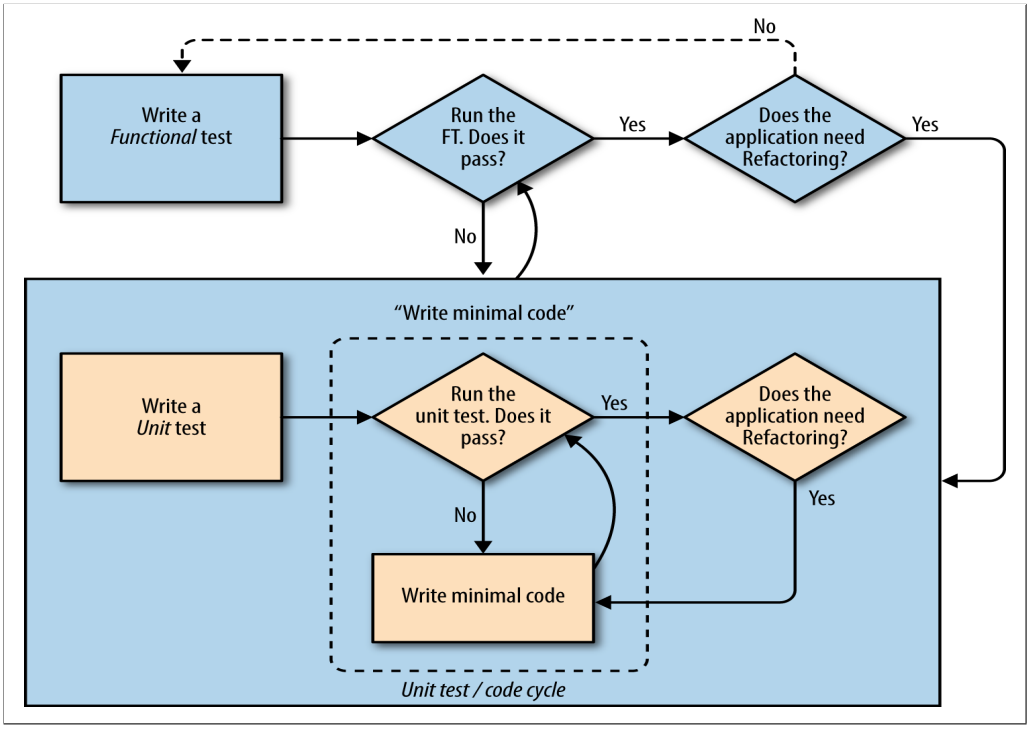


Figure 7.2: A Single Cycle of TDD, Obtained From [108]

mal code is developed until all the unit tests have passed. Having passed all the tests, it checks whether the functional test has passed and decides whether the expected behaviour of the software is preserved or more functional test is required. Depending on the outcome of this decision, the process finishes or moves to another cycle.

In order to use TDD efficiently for all the algorithms, I designed an integrated test harness where all algorithms are executed independently with the same input and the required data is equally accessible for all algorithms. Having the test harness ready, I devised test cases according to the objectives determined in requirement analysis. These test cases mirror the key objectives of all the algorithms and correspond to the functionalities for which the implementation is initiated. The implementation is accomplished only if all the test cases pass successfully.

Apart from the test cases for core objectives, some test cases were also required to be able to validate the output of each algorithm. This validation was necessary – particularly for the three comparative algorithms because it ensured that the implementation had been done correctly. In order to perform such validation, I obtained the results of use-cases from the original papers where the three algorithms were experimented. Then I devised test cases according to these results and implemented the algorithms to pass these test cases. In the next two sections, I expand the approach taken in every step of implementing the core functions. I first illustrate and discuss the architecture of the developed test harness. Then I describe the process of the implementation. Finally, I demonstrate the steps taken in order to validate the outcome of implementation.

7.3.1 Test Harness

One prerequisite for the experiment is to ensure that the comparative algorithms can run independently. This is because the high level of dependency between algorithms complicates the process of assessment and validation. In order to achieve this, the algorithms must have independent access to the required data prior and during the run-time. This is one of the objectives of a *test harness* which rose in the early 1970's for the purpose of integrated testing [104]. Test harness is commonly used for assessing the behaviour of modules with different sizes and natures during their interactive process. It is specific to a development environment and is built and configured according to the requirements [72].

The testing tools offered in Visual Studio cover all the required abilities of a test harness including automated and integrated testing. Hence in the case of my study, I was only required to design an appropriate solution with three goals. The first goal was to ensure the required data was accessible by all the four algorithms from a single source. The second goal was to ensure the required data for each algorithm was accessible

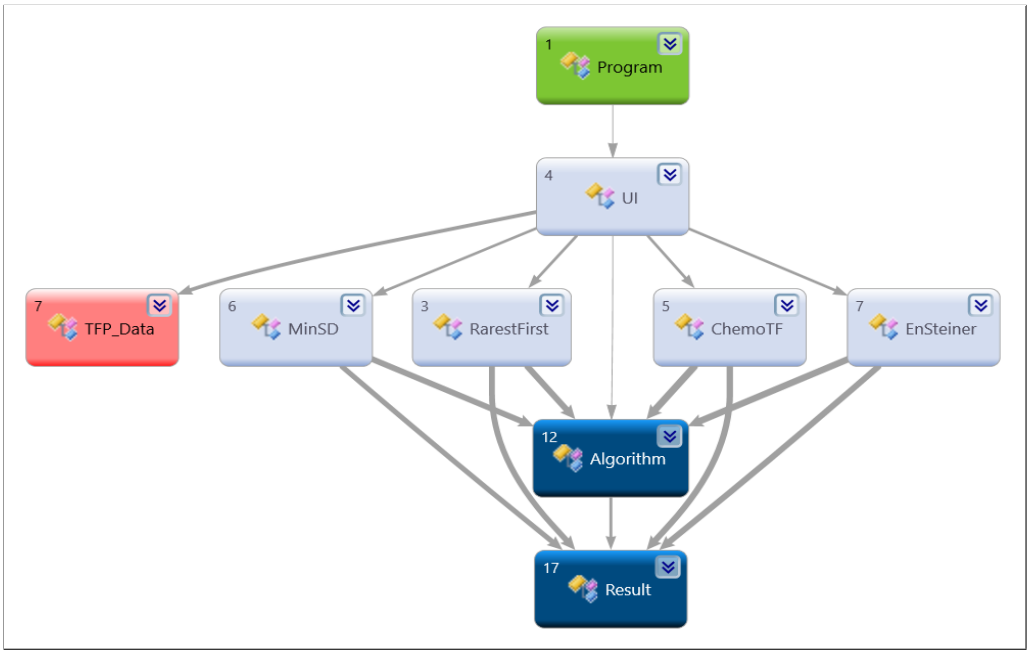


Figure 7.3: Class Dependecy Diagram of implemented solution

during the execution. The final goal was to guarantee that the execution of all four algorithms was independent and could not be affected or interrupted by any process. With the three above mentioned goals in mind, I designed a platform for implementing the algorithms. Figure 7.3 illustrates the dependency of each class corresponding to algorithms.

As observed from this figure, *Program* (shown by green colour on the top) is initiated from a single point by calling *User Interface* (UI) class. UI has dependency with five other classes including the algorithms and a class called *TFP Data*. TFP Data class (highlighted by red colour) contains all the required data for the algorithms. This data includes the social network and its all components such as experts, skills, areas of expertise, experiences, etc. This class has only one dependency with UI class, meaning that (1) the data is available from one source and (2) the data is available upon the execution and remains accessible during the execution for all the algorithms. This description satisfies the first two design

goals elaborated in the previous paragraph.

Furthermore, it can be observed that there is no dependency between the classes corresponding to the algorithms. Apart from the UI class, the *Algorithm* and *Result* (highlighted by dark blue colour) are the only point of dependency for the algorithms. This is because all the algorithms are defined in separate classes while inheriting common characteristics of Algorithm and Result classes. In other words, the Algorithm and Result classes reflect the similarities between algorithms whereas each algorithm defines its own dissimilarity. Such a design suggests that (3) the algorithms run independently without any deflections which satisfies the last goal discussed earlier.

In summary, I designed a platform for reliable execution of algorithms. In the next section I discuss the process of implementing key functionalities of the algorithms using TDD.

7.3.2 Implementing Core Functions

I began the process of development by designing test cases for the the algorithms. The ultimate goal of all four algorithms is to find the best team of experts according to their own criteria. Given that the metrics employed in each algorithm are calculated differently, a separate test unit for each algorithm was developed to ensure the results of the calculations were accurate. Figure 7.4 demonstrates a snapshot of the designed test cases which were passed by implemented functions.

The first algorithm I designed test cases for and implemented accordingly was ChemoTF. Given that this algorithm uses four metrics to form the final team, the validity of each metric is an essential requirement. The aim of implementation in this case was to turn the mathematical definition of the metrics discussed in Chapter3 into functions that could pass the corresponding test case. These metrics include *Communication Cost*, *Chemistry Level*, *Expertise Level*, *Expert Cost*, *Expert Cost*. Thus I devised

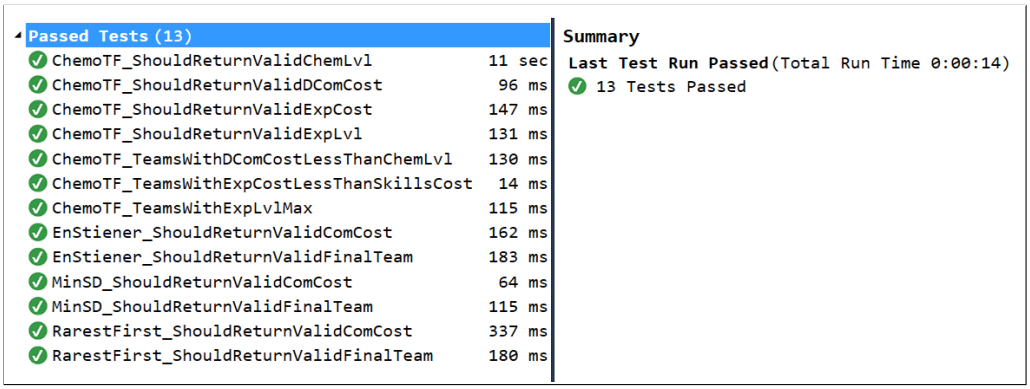


Figure 7.4: An snapshot of test cases passed for implementation

a set of test cases to ensure that the outcome of each corresponding function was as expected. The test cases for these metrics have been given in the first four lines of the Figure 7.4. The next three use-cases in this figure ensure that the comparison between various metrics are implemented correctly. The implementation in this case covered the mechanism employed in ChemoTF.

I implemented the core functions of comparative algorithms namely *EnSteiner*, *MinSD*, and *RarestFirst*. Communication Cost is the only metric these algorithms use while forming the final team thus developing corresponding functions to calculate this metric was the the main requirement. Since each algorithm calculates Communication Cost differently, I devised separate test cases for each algorithm. These test cases have been demonstrated in Figure 7.4 and are recognisable by their name which ends with "_ShouldReturnValidComCost". In addition, given that each one of the comparative algorithms employ a different mechanisms to form teams, extra test cases were required to guarantee the validity of final teams generated by each algorithm. These cases have been demonstrated in Figure 7.4 for each one of the comparative algorithms and are recognisable by their names which end with "_ShouldReturnValidFinalTeam". The next section describes the steps taken to pass these use-cases and validate the results of the comparative algorithms.

Table 7.1: The result of three comparative algorithms used as test cases

Algorithms	Required Skills			
	Diversity Dissmilarity Coverage	*Unified Ranking Probablistic Database*	*Multilingual Interface Ranking Constraints*	*Subgraph Maintenance Streaming Real-time Identification*
Ensteiner	E. Pitoura, C. Faloutsos	A. Deshpande	A. Kumaran, H. Jagadish, M. Ramanath, W. Fan	P. Indyk
MinSD	R. Agrawal	A. Deshpande N. Koudas J. Yang	A. Kumaran, R. Agrawal, K. Shim	R. Motwani, Lakshmanan
RarestFirst	H. V. Jagadish, R. Agrawal	A. Deshpande, R. Motwani, J. Widom	A. Kumaran, H. Jagadish, M. Ramanath, W. Fan	P. Indyk, N. Koudas

7.3.3 Validating the Outcome

In order to guarantee that my implementation of comparative algorithms reflects the exact mechanism employed in each algorithm and returns valid results, I obtained the experimental results of use-cases for each algorithm from their original papers. I then verified the results to the comparative study conducted in [130] and devised a list of use-cases to cover all the possible scenarios. Table 7.4 demonstrates the use-cases and the generated results for each one of the comparative algorithms. I then turned these use-cases and their corresponding results into test cases and devised functional test cases. Passing these test promise that my implementation returns the expected results as obtained in the original papers.

In summary, I implemented the core functionality required for all al-

gorithms in an accurate and reliable fashion. I first prepared a test harness to guarantee that the implementation was fair and accurate. Then I devised test cases for each algorithm according to the core requirements listed earlier in requirement analysis. I turned the parameters and the logic behind the mechanisms into test cases and programmed a software to pass the tests. I verified the mechanism and results of each algorithm against the experimental results obtained from the original papers by implementing such functions that passed devised test cases.

Using TDD helps to start the project rapidly and accurately without any consequent evaluation or validation. However, as mentioned earlier, the nature of TDD demands minimal code to only fulfil the core requirement. This is why TDD is usually equipped with Object Oriented Programming (OOP) to help develop the core functions implemented in the beginning and expand them into complete software. The next section discusses how I used this approach to develop a reliable software for the four mentioned algorithms.

7.4 Implementation: An OOP approach

Object-Oriented Programming (OOP) is a software programming technique based on the concept of "objects", which are data structures containing both functions (often refereed to as methods) and data (often refereed to as fields, attributes or properties). [106]. Objects are building blocks of OOP and are the main point of difference between OOP and procedural programming. They have characteristics of their existence (states); they are informative of their functionality (methods); they hold information (attributes); they live and die (instantiated or disposed); and they have definitions (class).

OOP have numerous advantages which have made it favourable since its emergence in mid 1980's. The characteristics such as robustness, extendability, compatibility, efficiency, portability, ease of use, functional-

ity, timeliness, inheritance, memory management, etc. are all advantages of using OOP [106]. However the last key benefits of OOP mentioned above (inheritance and memory management) were the two main reasons I adopted this technique for the purpose of my implementation.

Being able to take advantage of the concept of inheritance, I implemented similar attributes differently for all the algorithms. An evident example of these attributes is Communication Cost which is calculated differently across the algorithms. I devised a base class containing the similar attributes and methods and have each algorithm as a child class inheriting these attributes and methods. The algorithms then override the inherited attributes and methods according to their own definition and interest. Furthermore, the memory management capability offered as *Garbage Collection* (GC) in OOP made it easy to take advantage of the hash tables. Recall from Chapter 3 that the use of hash tables reduced the time complexity of generating sub-graphs of experts to linear time complexity. These hash tables reside in the memory and can grow big for large social networks. Taking advantage of GC and defining the classes accurately, each object is disposed as soon as its process is finished.

The rest of this chapter has been dedicated to explaining the steps taken for implementation of all the required features of the algorithms. These steps include devising a class to make the data available for algorithms during their execution, and implementing classes containing the attributes and methods for each algorithm.

7.4.1 Mapping Data

In order to make the required data available for the algorithms, I developed a base class called *ClassLoadable* consisting of a field called *dataRow*. The aim was to create a capability in which any instantiated object of a class that inherits ClassLoadable can access data directly from database in the form of dataRow. Then, I developed 8 classes corresponding to

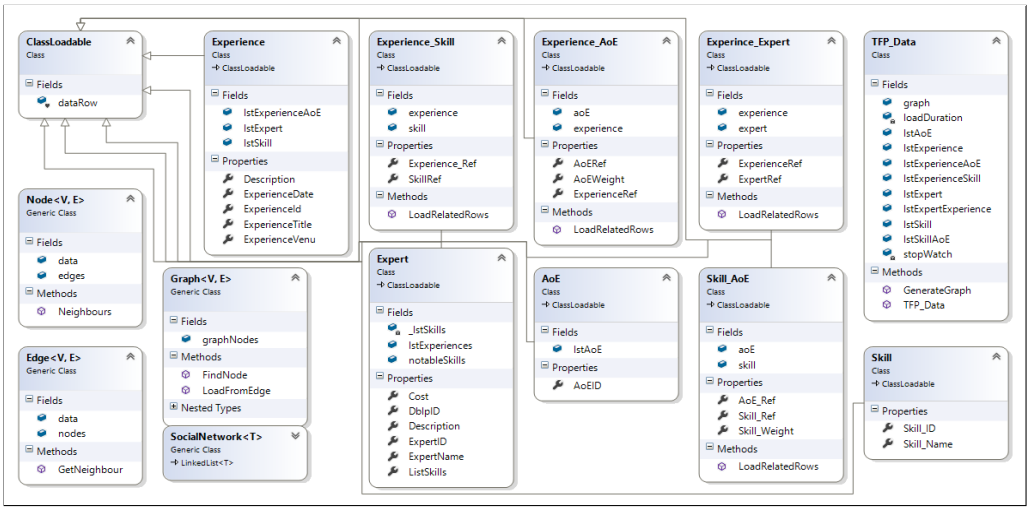


Figure 7.5: UML Class Diagram of Data Mapping Design

the 8 tables of the database in Figure 7.1, such that they all inherit ClassLoadable. Such design enables an efficient data mapping and provides a unified data platform for all the algorithms. Figure 7.5 illustrates the UML class diagram of this design. ClassLoadble depicted in the top right corner of the figure is the base class. The 8 classes which inherits this class are *Experience*, *Experience_Skill*, *Exert*, *Experience_AoE*, *AoE*, *Experience_Expert*, *Skill_AoE*, and *Skill*. They access the database directly and make the required data available by mapping it into *TFPData* demonstrated on the right side of the figure. The arrows from these classes to ClassLoadable demonstrates their dependence.

Apart from TFPData, there are other classes that have no dependency to the base class. These classes includes *Node*, *Edge*, *Graph*, and *SocialNetwork* and have been demonstrated on the left bottom of the Figure 7.5. They are used to construct the social network and populate the hash table containing sub-graphs of experts prior to the execution. Both social network and hash table reside in the memory during the run-time hence they are accommodated upon the initiation of the programme.

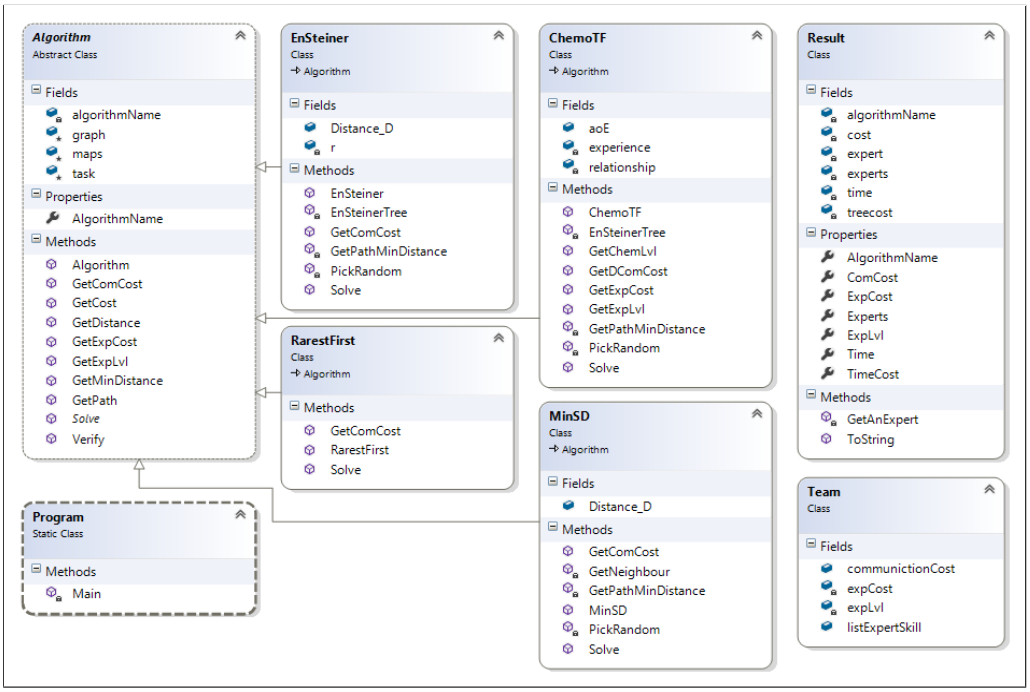


Figure 7.6: UML Class Diagram of Algorithms Design

7.4.2 Developing Algorithms

Having mapped the data into classes and constructed the necessary component of the application, I expanded the development of key functionalities of algorithms which previously implemented using TDD. In order to achieve this, I designed an *abstract* base class called *Algorithm* that is inherited by four classes corresponding to four algorithms in this. Figure 7.6 illustrates the UML class diagram of this design. These classes include *EnSteiner*, *RarestFirst*, *ChemoTF* and *MinSD*. The arrows from these classes to the base class specify the dependency.

The Algorithm class is an abstract class meaning that it contains no implementation and only indicates the common attributes necessary for a TF algorithm. In other words, any class that inherits it can implement the indicated attributes according to its own requirements. Each algorithm has their own class and implementations. The common attribute

between all the algorithms is the method *Solve*. This method executes the algorithm and is the only method that is called from *Program* class. Program class depicted on the left bottom of the figure is the only static class throughout the application. This is because it is the point where the application is initiated. Furthermore, *Result* and *Team* classes shown in the right side of the figure contain the results of the execution and are instantiated for every algorithm upon finding the desired team. The data in these classes is reported and stored into files as soon as the process in all algorithms are finished. The programme is terminated upon obtaining the results.

To summarise this chapter, I implemented a reusable, expandable and efficient software to produce reliable results for all the four algorithms. TDD was used to implement the core objectives identified in requirement analysis and validate the results. This process relies on the data offered from a single source thanks to the test harness. Then OOP was used expand the implementation to cover all functionality of the software. Thanks to the concepts of inheritance, the algorithms run independently with the same input under the same conditions. The following chapter provides an analysis based on the experimental results of my implemented software for all the algorithms discussed in this thesis. The experiment is conducted for tasks with different size and nature and is repeated to conclude a reliable analysis. First the metrics for the analysis is outlined. Then an analysis based on the results of ChemoTF is given. Moreover, a comparative analysis between all four algorithms based on the experimental results is provided. Finally sensitivity analysis for the employed metrics and parameters is presented.

"The fundamental principle of science, the definition almost, is this: the sole test of the validity of any idea is experiment."

Richard P. Feynman [52]

Chapter 8

Results and Analysis

Having designed, implemented, tested and validated all the algorithms and their corresponding entities in the previous chapter, an experiment was conducted on the CompScholarCorp. Evaluating the result of this experiment in real-life scenarios requires measuring the success rate of various tasks performed by teams consisting of real people who are formed by each algorithm. This necessitates an overwhelming amount of time and budget which makes a real-life experiment and the consequent evaluations impractical. Instead, I conducted a comparative analysis to evaluate my results against three TF algorithm with respect to well-known metrics. In addition I conducted a sensitivity study to determine the effect of inputs on the parameters and the effect of metrics on each other.

First the environment of the experiment and it configurations are discussed. In addition, the metrics used for the analyses are elaborated and the characteristics of each metric are explained. Then in the second section, the experimental results of ChemoTF are analysed and the quality of teams formed by my algorithm with respect to its parameters are assesses. In the third section, the results of comparative study for the four algorithms with respect to the selected well-known metrics are explained and the findings are outlined. Finally the results of the sensitivity study are elaborated and the findings are summarised.

8.1 Experiment Configurations

The experiment was conducted using a single machine. The machine used for this purpose was equipped with an Intel® Core™ i5-7300HQ CPU with 6 MB of cash and 3.50 GHz frequency, 8 GB of RAM, and 1 TB local hard disk. This machine was running Windows 10 Enterprise Edition as Operating System. The database and the software were copied into the local drive of the machine prior to the experiment. A 2% CPU usage, 7.1 GB available RAM, and 420 GB space in the local dive was detected before the experiment. In addition, the machine was connected to LAN due to the university policies.

The input generation procedure was adjusted to randomly generate tasks with p number of skills $p \in \{2,3,\ldots,20\}$. For each configuration, the task generation is repeated 100 times and the results were reported accordingly. This adjustment is based on the pattern followed by the original authors of the comparative algorithms. The results of the experiment were stored in a local *CSV* file for each algorithm. The rest of this chapter has been dedicated to discussing the metrics used for the analysis and explicating the result of the experiment with respect to this metrics.

8.2 Analysis Metrics

For the purpose of this study two sets of metrics are used to analyse the experimental results. The first set of metrics include the parameters introduced and defined earlier in Chapter 3 for ChemoTF. These parameters include, *Chemistry Level*, *Dynamic Communication Cost*, *Expertise Level*, and *Expert Cost*. These parameters are used as metrics to analyse the experimental results of ChemoTF in order to provide an insight of the quality of the generated teams. The best team with respect to these parameters is the one which has Communication Cost lower than Chemistry Level, the highest Expertise Level, and Expert Cost lower than the

average cost of the teams possessing the same skills in social network. Among these parameters, Chemistry Level and Dynamic Communication Cost are unique to ChemoTF and cannot be used to analyse the results of other algorithms. However Expertise Level and Expert Cost are compatible with other algorithms thus they have been used for the purpose of comparative analysis.

The second set of metrics used in this study are the well-known metrics commonly used in the literature to assess the quality of a successful team. These metrics include the *cardinality*, *sum of degree centrality*, and *density*. Since the execution time is an essential factor of of any algorithm, the *performance* of each algorithm with respect to time (seconds) has been considered as a metric as well. The above mentioned four metrics have been employed in the comparative analysis as well as the sensitivity analysis conducted in this chapter. The intuition behind using these metrics is to provide a fair comparison between ChemoTF and the three algorithms including *RarestFirst*, *EnSteiner*, and *MinSD*. In order to understand the notion of these metrics and the rationale behind choosing them for my analysis, the rest of this section has been dedicated to providing a background on each metric and describing their characteristics.

8.2.1 Cardinality: The smaller, the better

The first metric of this study is the cardinality of final team which is referred to as the number of vertices in a graph. The consensus among researchers is that teams with smaller cardinality are more desirable. Teams with lower number of experts are expected to be more cost-effective. A huge part of the budget for every project is assigned to cover the personnel cost and coordination [7], hence teams with small size have been the target for many algorithms in the area of TF. In addition it is suggested that teams with smaller cardinality have higher level of productivity [24] and are more effective when it comes to decision making [23].

8.2.2 Centrality: The higher, the better

The second metric is the centrality which determines the most significant vertices within a graph [25]. Historically, the most basic centrality measure for undirected graph is the *degree centrality* which is determined by the number of neighbours for a vertex. The degree centrality is first proposed and formulated by Freeman [55] and has been used since in various fields including social networks. For the purpose of this analysis, this formula has been used to calculate the centrality. I have referred to degree centrality simply as "centrality" in this chapter. The consensus among researchers is that the more the centrality for a vertex gets, the more influence that vertex becomes. [26]. Base on this agreement, teams with higher centrality are more desirable.

8.2.3 Density: The higher, the better

Density is the third metric I employed for the purposed of analysis in this study. As described earlier in Chapter2, the density of a graph determines the sum of the edges to all its vertices [37]. The motivation behind using the density as a metric is that the teams with higher density are more interconnected hence their communication channels are abundant. In the area of TF, this metric has been used in various studies including [57], [119],and [109]. The consensus is that higher density is desirable for the teams.

8.2.4 Performance: The faster, the better

The final metric used in this analysis is performance of algorithms with respect to time. The consensus is that the lower the time of execution for an algorithm is, the better the performance of that algorithm becomes. However, this measure does not solely represent the quality of an algorithm for it relies on the process rather than the results.

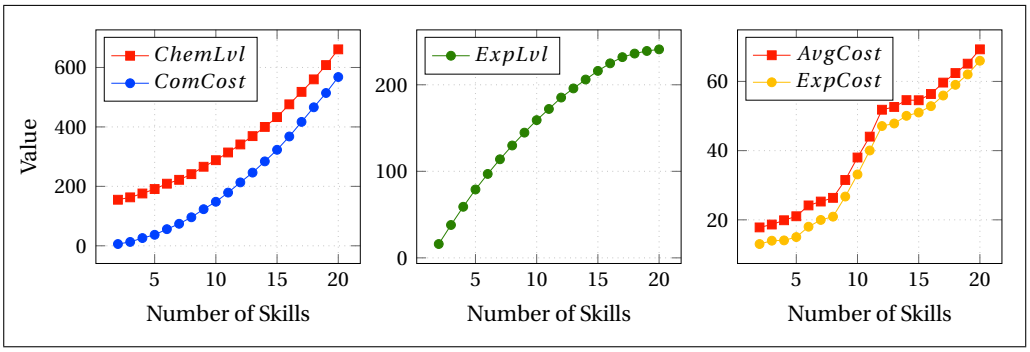


Figure 8.1: The effect of the number of skills on the final teams formed by ChemoTF with respect to its parameters

In summary, three analyses are conducted using two sets of metrics including 4 metrics from ChemoTF and 4 metrics commonly used in the literature. These metrics are Communication Cost and Chemistry Level for ChemoTF only, Time performance for comparative analysis only, and Expertise Level, Expert Cost, cardinality, centrality, and density for both comparative and sensitivity analysis. The rest of this chapter has been dedicated to elaborating these analyses and discuss the findings.

8.3 ChemoTF Results and Analysis

In this section, I present the experimental results of ChemoTF and analyse the final teams with respect to five parameters employed in ChemoTF given various number of required skills. The effect of number of required skills in the task on the final teams generated by ChemoTF given various metrics has been illustrated in Figure 8.1. These metrics include Chemistry Level ($ChemLvl$), Communication Cost ($ComCost$), Expertise Level given the skills ($ExpLvl$), Average Cost of experts possessing the skills in social network ($AvgCost$), and Expertise Cost ($ExpCost$). Given the similar nature of $ChemLvl$ compared to $ComCost$, and $AvgCost$ compared to $ExpCost$, these metrics have been studied together.

8.3.1 Communication Cost and Chemistry Level

The Communication Cost ($ComCost$) and Chemistry Level ($ChemLvl$) of teams formed by ChemoTF have been demonstrated on the left side of the Figure 8.1. As observed from this figure, ChemoTF always finds teams with $ComCost$ less than $ChemLvl$. The difference between the values of these function is 163 at the highest, 87 at the lowest, and 128 as median average. This is because the mechanism of ChemoTF has been designed such that the Dynamic Communication Cost of a pair of experts given their skills is compared to the Chemistry Level of skills in each round. Pairs for which the Dynamic Communication Cost exceeds the Chemistry Level are ignored which results in teams with $ComCost$ less than $ChemLvl$. Given that $ChemLvl$ represents the required level of communication for a team to succeed, this observation can be summarised as the following statement:

Statement 8.3.1. ***(ChemoTF Communication Effectiveness)***

Teams formed by ChemoTF communicate effectively with the rate of 100% *of the required communication level.*

It is also observed that both $ComCost$ and $ChemLvl$ increase exponentially as the number of skills grows. Both functions are fit for $y = ne^{mx}$ function. The goodness of fit for $ChemLvl$ with values $n = 127.25$ and $m = 0.0821$ is calculated as $R^2 = 0.99$. The goodness of fit for $ComCost$ with values $6.58 \leq n \leq 15.67$ and $0.19 \leq m \leq 0.25$ is $R^2 = 0.91$. This suggests that the growth of both functions depends on the number of the skills. The exponential growth rate for $ComCost$ is $k = 0.369$ and for $ChemLvl$ is $k = 0.323$. The reason for such behaviour is that an increase in the number of skills results in larger teams and consequently a decline in the number of communication channels. This forces ChemoTF to form more diverse teams with relatively higher $ComCost$. The impact of the number of skills on $ComCost$ is calculated at the rate of 36%. The statement bellow summarises this observation:

Statement 8.3.2. ***(Impact of Number of Skills on Communication Cost)*** *The growth of the number of skills has a negative impact of* 36% *on the level of communication in ChemoTF formed teams.*

Furthermore, $ChemLvl$ and $ComCost$ gradually converge. The distance between two functions at the beginning is 163 and decreases to 87 at the end. This is due to the reduction of commonality between the individual skills which leads to less overall $ChemLvl$. ChemoTF uses this opportunity to form teams with high overall expertise level whose members have weaker social ties in the network which results in higher $ComCost$ in overall. Having said that, the two functions will never intersect due to the $ChemLvl$ limit. This observation can be stated as bellow:

Statement 8.3.3. ***(Communication Cost with Large Number of Skills)*** *Teams formed by ChemoTF that cover tasks larger than 20 skills are likely to communicate at rate of* 90% *of Chemistry Level.*

Finally, the curves in both functions suggest that when the number of required skill is low, ChemoTF finds smaller teams with high level of expertise and less $ComCost$. The possibility of finding more experts with multiple skills is higher compared to when the number of required skills are high. This behaviour causes near-mirror curve in $ExpLvl$ demonstrated on the middle graph of the figure. This is particularly accurate for tasks with 5 to 15 skills. The dissimilarity of $ExpLvl$ and $ComCost$ in this area is calculated using *Kolmogorov-Smirnov* (KS) test at its lowest rate of $D_{KS} = 189.33$. This rate is 53% lower compared to smaller tasks and 76% lower compared to larger tasks which suggests:

Statement 8.3.4. ***(Task with 5 to 15 Skills in ChemoTF)*** *Teams formed by ChemoTF for covering 5 to 15 skills have Low Communication Cost and high Expertise Level with the best median average ratio of* $\frac{ExpLv}{ComCost} = 1.04$ *which is* 53% *better compared to smaller tasks and* 76% *better compared to larger tasks.*

Having analysed the results of ChemoTF with respect to Chemistry Level and Communication Cost, I conclude that ChemoTF performs as expected with regards to efficient Communication Cost. In the next section I elaborate the results of ChemoTF with respect to Expertise Level.

8.3.2 Expertise Level

The result of ChemoTF with respect to Expertise Level ($ExpLvl$) of the team given various number of skills have been demonstrated in the middle graph of Figure 8.1. As stated in Chapter 3, one objective of ChemoTF is to find teams with maximum possible $ExpLvl$ hence no limitation has been set for this metric.

As suggested in previous section, the curve in $ExpLvl$ is the result of high number of multi-skilled experts with high $ExpLvl$. In addition to this curve, it can be observed that $ExpLvl$ experiences logarithmic growth as the number of required skills rises. The function $y = n\ln x - m$ with values $107.54 \leq n \leq 111.64$ and $84.129 \leq m \leq 90.156$ is 98% fit for $ExpLvl$ function. The growth of this function at the beginning is calculated at the rate of $k = 1.01$ which drops to $k = 0.34$ for larger tasks with 20 skills. The reason for this behaviour is that $ExpLvl$ becomes unified across the network. More specifically, the average sum of $ExpLvl$ of experts who can communicate efficiently and fulfil the task becomes very similar across the network. This is governed by $ChemLvl$ to guarantee the communication efficiency. As a rule of thump, this observation can be stated as bellow:

Statement 8.3.5. ***(Normalisation of Expertise Level in ChemoTF)***

Teams formed by ChemoTF that cover tasks larger than 20 skills are 98% *likely to have similar overall expertise level of* $ExpLvl_{20} \pm \varepsilon$ *where* $ExpLvl_{20}$ *is the Expertise Level for 20 skills and* $\varepsilon < 4.23$.

As described earlier in Chapter 3, finding experts with maximum Expertise Level can potentially result in very costly teams. In order to control this cost, ChemoTF has been equipped with a mechanism to control

the Expert Cost. This function is referred to as Average Cost and is calculated for a team as the sum of the average cost of experts in social network with the required skills. The following section discusses the experimental results with respect to Expert Cost and Average Cost.

8.3.3 Expert Cost and Average Cost

The Average Cost ($AvgCost$) of skills in the social network and the Expert Cost ($ExpCost$) of the team given required skills have been represented on the right graph in the Figure 8.1. $AvgCost$ has a goodness of fit of $R^2 = 0.95$ with the polynomial function $y = -0.056x^2 + 4.52x$. The goodness of fit for $ExpCost$ is $R^2 = 0.96$ with the polynomial function $y = nx^2 + mx$ where $-0.012 \leq n \leq 0.008$ $(n \neq 0)$and $3.121 \leq m \leq 3.728$. This suggest a similar behaviour between the two functions which is proven by $D_{KS} = 27.45$ and is observed by their polynomial growth as the number of skills rises.

ChemoTF has been designed such that if $ExpCost$ of a selected candidate is found more costly than $AvgCost$ of experts possessing the same skills throughout the network, the candidate is substituted with another expert until a candidate with a $ExpCost$ lower than $AvgCost$ is found. This leads to a consistently lower overall $ExpCost$ of the team and shows that teams formed by ChemoTF are always less costly than the $AvgCost$. Considering that $AvgCost$ represents the required cost of a selected expert and $ExpCost$ is never above the threshold set by $AvgCost$, the statement bellow is given:

Statement 8.3.6. ***(ChemoTF Cost-Effectiveness)***

Teams formed by ChemoTF are 100% *cost-efficient with respect to the average cost of experts possessing the same skills in social network.*

Furthermore, there is a positive correlation between the two functions $ExpCost$ and $ExpLvl$ which is calculated using *Pearson Correlation Coefficient* at the rate of $r = 0.95$. Given the positive growth of $ExpLvl$ discussed earlier, the values of $ExpCost$ gets to the highest allowed values fil-

tered by $AvgCost$ function. This correlation reveals interesting findings. One finding is regarding the polynomial growth of $ExpCost$ between 5 to 15 skills. $ExpCost$ has a linear growth of the rate of 0.33 in the beginning and then turn to polynomial with a growth rate of 1.21 between 5 to 15 skills. This is the same area where a near-mirror is detected in $ExpLvl$ function. Teams in this area have high level of $ExpLvl$. Given the positive correlation between $ExpLvl$ and $ExpCost$, this phenomena is interpreted as a reaction to rise of $ExpLvl$.

Finally, $ExpCost$ and $AvgCost$ stay close to each other throughout the experiment. The median average difference between these functions is 4.55. Given that $AvgCost$ is the upper bound for teams formed by my algorithm, the closeness of these two functions suggests that ChemoTF forms teams near the upper limit of cost. The difference is higher in the beginning but two functions converge as the number of skills grows. The highest difference is 6.153 for task with 6 skills, and the lowest difference is 3.10 for task with 19 skills. The reason is that the cardinality of teams for small tasks is small due to high number of multi-skilled experts. The statement bellow can be given:

Statement 8.3.7. ***(Maximum Expertise Level in ChemoTF)***

Teams formed by ChemoTF have the highest Expertise Level among the teams filtered by the average cost constrain of similar teams.

In summary, analysing the experimental results of ChemoTF with respect of various metrics demonstrates that this algorithm forms teams in which members can communicate effectively within the range of required communication level. This level of communication is particularly effective for small to large tasks and reaches the highest boundary for extra large tasks. On the other hand, the teams formed by ChemoTF possess maximum possible level of expertise while being cost-effective. The cost of the team never exceeds the average cost of similar teams in the social network thus their cost-effectiveness is guaranteed.

In order to be able to validate the results of ChemoTF, I ran a comparative analysis and evaluated the results against three well-known algorithms. The experiment was conducted in a fair environment with the same inputs for all algorithms. The next section has been devoted to analysing the outcome of this comparative analysis.

8.4 Comparative Analysis

To determine the quality of the teams formed by ChemoTF, I evaluated my algorithm against three prominent state-of-the-art team formation algorithms; *RarestFirst* [86], *EnSteiner* [86], and *MinSD* [76]. My implementations of these algorithms were verified against the use-cases provided in the source papers and tested in a uniform context in the previous chapter. The metrics used for this analysis include *Expertise Level*, *Expert Cost*, *cardinality*, *sum of degree centrality*, and *density*. Given the different definitions of Communication Cost function across algorithms, this parameter has been removed from the comparative analysis. Figure 8.2 illustrates the result of the experiment with respect to these metrics. The left panel (1) depicts the distribution box plots and the graphs on right panel (2) gives the median average trends for four algorithms.

The first observation I make is that the results of ChemoTF (shown in green) are less variable than the other algorithms. Examination of the box plots in Figure 8.2(1) reveals that the lower and upper extremes of ChemoTF are very close to the median values in almost every case across the tasks – regardless of their size and complexity. With respect to $ExpLvl$, the median average distance between lower and upper extremes in my algorithm is 2% lower than RarestFirst, 58% lower than EnSteiner, and 33% lower than MinSD. This distance with respect to $ExpCost$, is 64% lower than RarestFirst, 59% lower than EnSteiner, and 41% lower than MinSD. This distance with respect to the other metrics is also consistently lower than all three algorithms. This is a significant achievement for ChemoTF

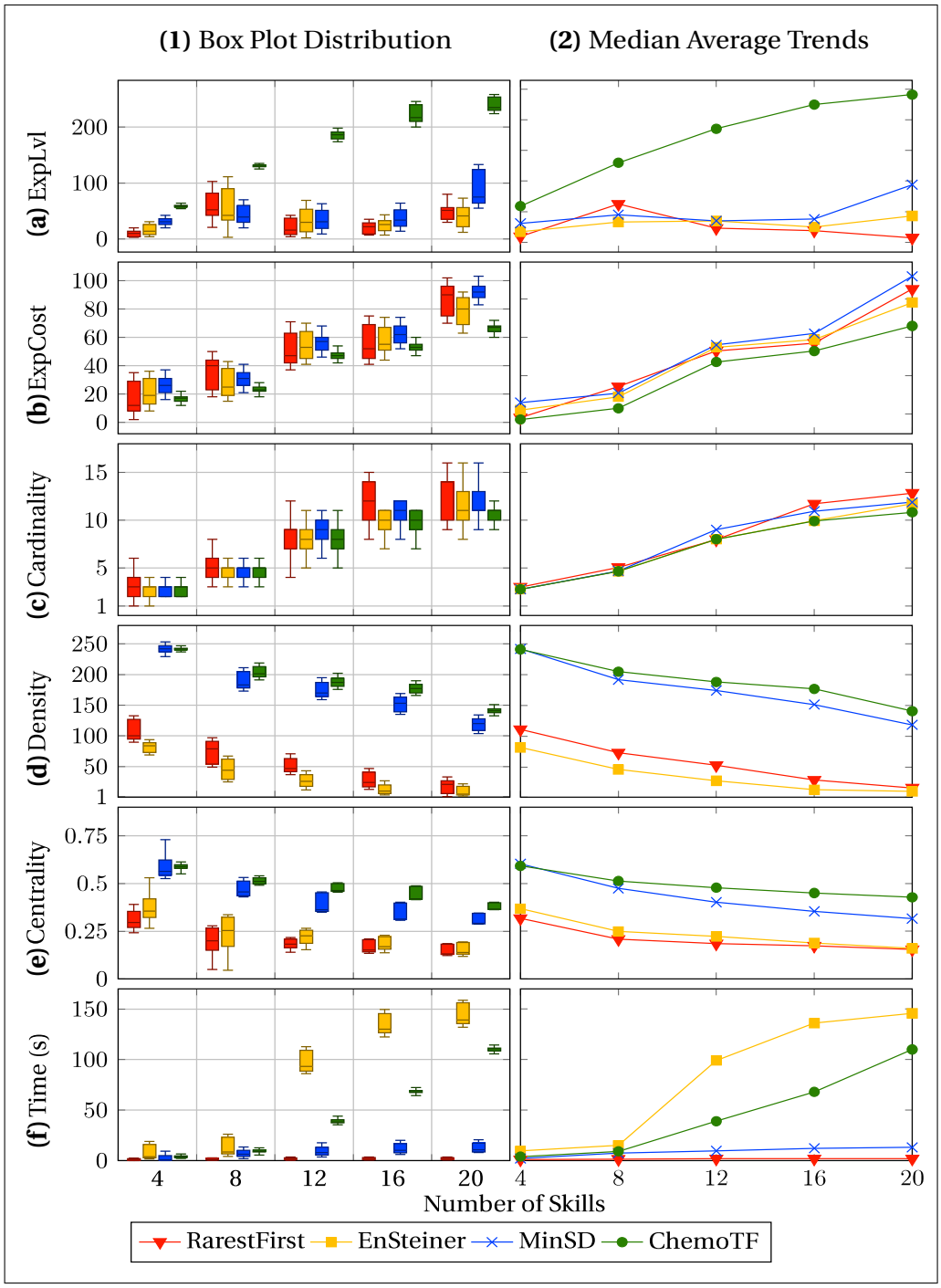


Figure 8.2: Comparison between experimental results of TF algorithms with respect to various metrics. The graphs on left demonstrate distribution box plots and the graphs on right display the median average trends.

particularly because of the fact that TF problem is NP hard. The statement bellow summarises this observations:

Statement 8.4.1. ***(ChemoTF Predictability)***
ChemoTF consistently generates more predictable results for TF problem compared to RarestFirst, EnSteiner, and MinSD with 61%, 68%, 36% *more accuracy.*

Apart from the predictability, the results suggest that ChemoTF performs better with respect to the metrics of the quality of a successful team. The rest of this section has been dedicated to conducting a comparative analysis on the all algorithms and providing an insight of the results of ChemoTF in a comparative manner.

8.4.1 Comparing Expertise Level of the Teams

The first metric used in this analysis is Expertise Level ($ExpLvl$). This metric is one of the main parameters of ChemoTF and has been fully discussed earlier in Chapter 3. Since it reflects the level of expertise of a team in a particular set of skills, teams with higher Expertise Level are expected to have higher potential to perform a task better.

As observed from row (a) in Figure 8.2, ChemoTF is exceptional with respect to $ExpLvl$. The results of $ExpLvl$ function for ChemoTF are 87.2% higher than RarestFirst, 82.1% higher than EnSteiner, and 81.12% higher than MinSD at their best, with respect to the highest achievable $ExpLvl$ values. In the worst case, these results are 52% higher than RarestFirst, 52.7% higher than EnSteiner, and 28.12% higher than MinSD. The dissimilarity of ExpLvl for ChemoTF using KS Test is calculated at the rate of 672.17 compared to RarestFirst, 706.98 compared to EnSteiner, and 597.14 compared to MinSD. These results demonstrate that ChemoTF is working as intended as it has been designed to prioritise $ExpLvl$. This the statement bellow can be given:

Statement 8.4.2. ***(The Most Expert Teams with ChemoTF)***

The teams formed by ChemoTF have the highest Expertise Level, 73% *higher than RarestFirst,* 72.3% *higher than EnSteiner, and* 56.66% *higher than MinSD at the median average rate with respect to the highest achievable Expertise Level.*

Given the consistency of ChemoTF in forming teams with highest Expertise Level for tasks with various complexities, this algorithm is the best choice when high expertise of the team members is a priority.

8.4.2 Comparing Expert Cost of the Teams

The second metrics I used for the purpose of this comparative analysis is Expert Cost ($ExpCost$). As described in Chapter 3, the less the value of $ExpCost$ gets for a team, the more cost-effective the team becomes.

The experimental results with respect to $ExpCost$ are given in row (b) of Figure 8.2. The results of $ExpCost$ for ChemoTF are 38.9% lower than RarestFirst, 36.1% lower than EnSteiner, and 45.8% lower than MinSD at their best rates with respect to average cost of team in the network. In addition, these results are 7.3% lower than RarestFirst, 10.16% lower than EnSteiner, and 14.78% lower than MinSD at their worst rates with respect to average team cost. Furthermore, ChemoTF is the only algorithm that achieves a success rate of 100% in forming teams with $ExpCost$ bellow the average cost. The success rates for RarestFirst, EnSteiner and MinSD are 11.5%, 9.8%, and 5.4% respectively. This is due to the cost function embedded in ChemoTF, whereas the other algorithms do not employ any mechanism to control the cost of the final teams. This observation can be included as the statement bellow:

Statement 8.4.3. ***(Lowest Expert Cost with ChemoTF)***

ChemoTF achieves forming teams with the lowest Expert Cost with the success rate of 100% *compared to RarestFirst, EnSteiner and MinSD with the success rate of* 11.5%, 9.8%, *and* 5.4% *respectively.*

In addition, it is evident that the $ExpCost$ for all algorithms increases as the number of skills grows. This is because, more experts are required to cover the skills as the number of skills grows hence the associated cost increases accordingly. Apart from ChemoTF which consistently forms cost-effective teams, other algorithms form teams with almost similar $ExpCost$. The dissimilarity rates between ChemoTF and RarestFirst, EnSteiner, and MinSD are 46.58, 41.91, and 65.89 respectively.

8.4.3 Comparing Cardinality of the Teams

As discussed earlier, a significant part of the budget for every project is assigned to cover personnel cost and coordination [7]. In fact it was found that the cardinality of the team has a positive correlation at the rate of 98.82% with $ExpCost$. Thus teams with small sizes are more desirable. As observed from row (c) of Figure 8.2, ChemoTF has the best performance with respect to this metric. The median average cardinality rate in ChemoTF is 27.8% of the best achievable ratio. This value for RarestFirst is 18.94%, for EnSteiner is 25.98% and for MinSD is 21.56% respectively. The reason lies in the low number of intermediate nodes in the teams formed by ChemoTF. This is achieved by $Neighbour$ function described earlier which leads to cohesive teams with small sizes. Thus I state that:

Statement 8.4.4. ***(Lowest Cardinality with ChemoTF)***

Teams formed by ChemoTF have the lowest cardinality among all the algorithms. The median average cardinality achieved in ChemoTF is 8.86% *lower than RarestFirst,* 1.82% *lower than EnSteiner, and* 6.24% *lower than MinSD with respect to the best achievable cardinality.*

The ability of ChemoTF to consistently form small teams for tasks with different types and complexities makes it the best choice for the scenarios where the budget is limited. In addition, given that small teams are easy to manage and more productive [74], ChemoTF stands out when the efficiency of management or coordination is crucial.

8.4.4 Comparing Density of the Teams

As discussed earlier, teams with higher density are associated with effective communication due to the abundance of communication channels. [37]. The experimental results with respect to the density shown on row (d) of Figure 8.2 suggests that ChemoTF achieves the density rates of 60.27%, 25.51%, 15.67%, 11.05%, and 7.04%, for the tasks with 4, 8, 12, 16, and 20 skills with respect to the density of the perfect graph. These rates are 32.51%, 16.35%, 11.26%, 9.24%, and 6.23% higher than RarestFirst, 39.79%, 19.74%, 13.36%, 10.23%, and 6.50% higher than EnSteiner, and finally −0.19%, 1.52%, 1.15%, 1.59%, and 1.11% higher than MinSD. The median average rate of the density for ChemoTF is 23.9% which is 15.12% higher than RarestFirst, 17.92% higher than EnSteiner, and 1.03% higher than MinSD.

The reason for the remarkable results achieved by ChemoTF is that teams formed by this algorithm take advantage of small number of mediators. The mediators are those experts that connect two or more experts to each other in social network and are chosen when the Communication Cost on the route through them is minimum. Given that ChemoTF does not aim at forming teams with minimum Communication Cost thanks to its novel parameters, it uses less mediators thus forms team with higher coherency and consequently higher density. To summarise this observation, it can be stated that:

Statement 8.4.5. ***(Highest Density with ChemoTF)***

Teams formed by ChemoTF have the highest density of a median average ratio of 23.9% *among all the comparative algorithms which is* 15.12% *higher than RarestFirst,* 17.92% *higher than EnSteiner, and* 1.03% *higher than MinSD respectively.*

The difference between the density of ChemoTF and MinSD is subtle for small tasks but becomes larger as the number of skills rises. The dissimilarity between these two algorithms starts at $D_{KS} = 41.1$ and rises to

$D_{KS} = 73.77$ at the end. This is because MinSD also forms coherent teams for small tasks thanks to its definition of $ComCost$ based on the shortest distance between neighbouring nodes. However as the number of skills grows and more experts are required for the team, the possibility of the distance between neighbouring nodes being the shortest path decreases and more intermediate nodes are employed. Compared to RarestFirst and EnSteiner, ChemoTF is significantly better with respect to the density of the team. The dissimilarity between ChemoTF and RarestFirst is $D_{KS} = 669.39$ whereas this value between ChemoTF and EnSteiner is $D_{KS} = 772.15$. The reason for this significant difference is that both the algorithms use an extensive number of intermediate connectors to reduce $ComCost$ as much as possible which reduces the coherency of the teams.

Finally, the density of teams decline for all the algorithms as the number of skills grows. The reason is the growth of the team cardinality according to the number of skills. As the cardinality of the team increases, more communication channels are required to maintain the density of the final team. This can be easily observed in the equation $D = \frac{2|E|}{|V|(|V|-1)}$ [37]. In order to maintain the level of density D for an increasing number of vertices $|V|$, it is required to maintain the number of edges $|E|$. Because neither the network nor the final teams are perfect graphs, the density declines as the size of the task grows.

8.4.5 Comparing Centrality of the Teams

As stated earlier in this chapter, sum of the degree centrality [55] is a well-know metric commonly used in social network analysis which is often referred to as *centrality*. Nodes with higher value of centrality are believed to have higher influence across the network [26]. Considering the high centrality values of teams formed by ChemoTF shown in row (e) of Figure 8.2, this algorithm is the best choice when teams with higher impact are preferred. ChemoTF achieves the degree ratios of 59.3%, 51.3%, 47.8%,

45.1%, and 38.2% for tasks with 4, 8, 12, 16, and 20 skills. The median average degree for ChemoTF is 48.34% which is 27.74% higher than RarestFirst, 24.54% higher than EnSteiner, and 5.36% higher than MinSD. Essentially, ChemoTF forms teams with the highest level of expertise possible. Given that the $ExpLvl$ of a team reflect the sum of the number of edges to all the nodes in the team (sum of degree), the formed team naturally expresses a high centrality value. In addition, the centrality values follow the same pattern observed for the density. For all the algorithms, both metrics similarly decline as the number of skills grows. The reason is the positive correlation between centrality and density which has been proven in [126] which I calculated at the rate of 0.98%. I conclude that:

Statement 8.4.6. ***(Highest Centrality with ChemoTF)***

Teams formed by ChemoTF have the highest sum of the degree centrality of the median average of 48.34% *which is* 27.74% *higher than RarestFirst,* 24.54% *higher than EnSteiner, and* 5.36% *higher than MinSD.*

8.4.6 Comparing Performance of the Algorithms

The experimental results with respect to performance are presented in row (f) of Figure 8.2. Although my experiments were executed up to 20 skills, scaling on this parameter has limited value in reality – as issuing tasks with a high number of required skills has a negative impact on the performance of the team [24]. In fact, Miller [100] suggests that an ideal size of the team should be 7 ± 2. In addition, Blenko *et al.* [23] prove that teams consisting of more than 7 people suffer from lack of decision effectiveness. Analysing the records of my corpus, I found that the median value and average number of skills used in publications throughout the corpus are 5 and 6.13. Considering keywords as skills and publications as the outcome of collaboration between experts, the performance of the algorithms should be ideally be considered for a maximum of 12 skills – as any task larger than this will result in teams with undesirable cardinality

(for all four algorithms).

Overall, the performance of ChemoTF for tasks with maximum 8 skills is comparable to the performance of the other algorithms. The median average performance of ChemoTF is 2.5 times faster than EnSteiner. My algorithm at its best case performs equally well compared to RarestFirst and 5.1 seconds slower at its worst case. This performance is 0.8 ± 0.2 seconds slower than MinSD in average. Although the performance of ChemoTF drops for 8 to 12 skills, it remains tractable for the entire process. This can be observed from the extremely low difference between lower and upper extremes on row (f).1 of the Figure 8.2 indicating notable execution time stability. It can be stated that:

Statement 8.4.7. ***(Ideal Performance of ChemoTF)***

ChemoTF forms teams in an ideal time-line and is and in a more predictable fashion compared to other algorithms, with a median average rate of 1.2, 15.4, *and* 5.6 *times better than RarestFirst, EnSteiner, and MinSD.*

In order to determine the impact of ChemoTF parameters on the metrics used in this study, a sensitivity analysis is conducted with respect to different tasks. The next sections elaborates the outcome of this analysis.

8.5 Sensitivity Analysis

Sensitivity analysis is referred to as the study of how uncertainty in the output of a model can be attributed to different sources of uncertainty in the input of the model [113]. It is often employed to determine how changes in inputs, definitions or parameters can improve the accuracy or robustness of the results. [112]. Given that the parameters for ChemoTF were introduced for the first time in this thesis, I conduct a sensitivity analysis to study the impact of the parameters and metrics on each other. The parameters include *Communication Cost* ($ComCost$), *Expertise Level*

($ExpLvl$), and *Expert Cost* ($ExpCost$). The metrics include the *cardinality*, *density*, and *centrality*.

Figure 8.3 depicts the sensitivity analysis result. Each graph represent three factor including a parameter, a metric and a task consisting of various skills. The graphs (a) to (e) illustrate the impact of $ComCost$, graphs (f) to (i) demonstrate the effect of $ExpLvl$, and graphs (j) to (l) depict the effect of $ExpCost$ on the three metrics mentioned above. Since $ExpLvl$ and $ExpCost$ were previously used as metrics for comparative analysis, the effect of $ComCost$ on these two parameters have been studied as well.

8.5.1 Effect of Communication Cost

The effect of $ComCost$ on the metrics has been illustrated in graphs (a) to (e) of Figure 8.3. As observed from graph (a), $ComCost$ has a positive correlation with $ExpLvl$ which is almost independent of skills. I calculated thus correlation at the rate of 79%. This suggest that as the $ComCost$ increases, the number of communication channels decreases and the experts become more individual. This is governed by Chemistry Level and enables ChemoTF to maximise $ExpLvl$ as much as possible. I state that:

Statement 8.5.1. ***(Effect of Communication Cost on Expertise Level)***
There is +79% correlation between Communication Cost and Expertise Level in ChemoTF which is independent of the number of skills.

The sensitivity between $ComCost$ and $ExpCost$ has been shown in graph (b) of Figure 8.3. There is also a positive correlation between these two parameters. This correlation has been calculated at the rate of 77% which is found to be almost independent of skills. The reason lies in the positive correlation between $ExpLvl$ and $ComCost$ shown in the graph (a) and the positive correlation between $ExpCost$ and $ExpLvl$ shown in the graph (f). As $ComCost$ rises, $ExpLvl$ rises as well which enforces $ExpCost$ to rise accordingly. The statement bellow summarises this observation.

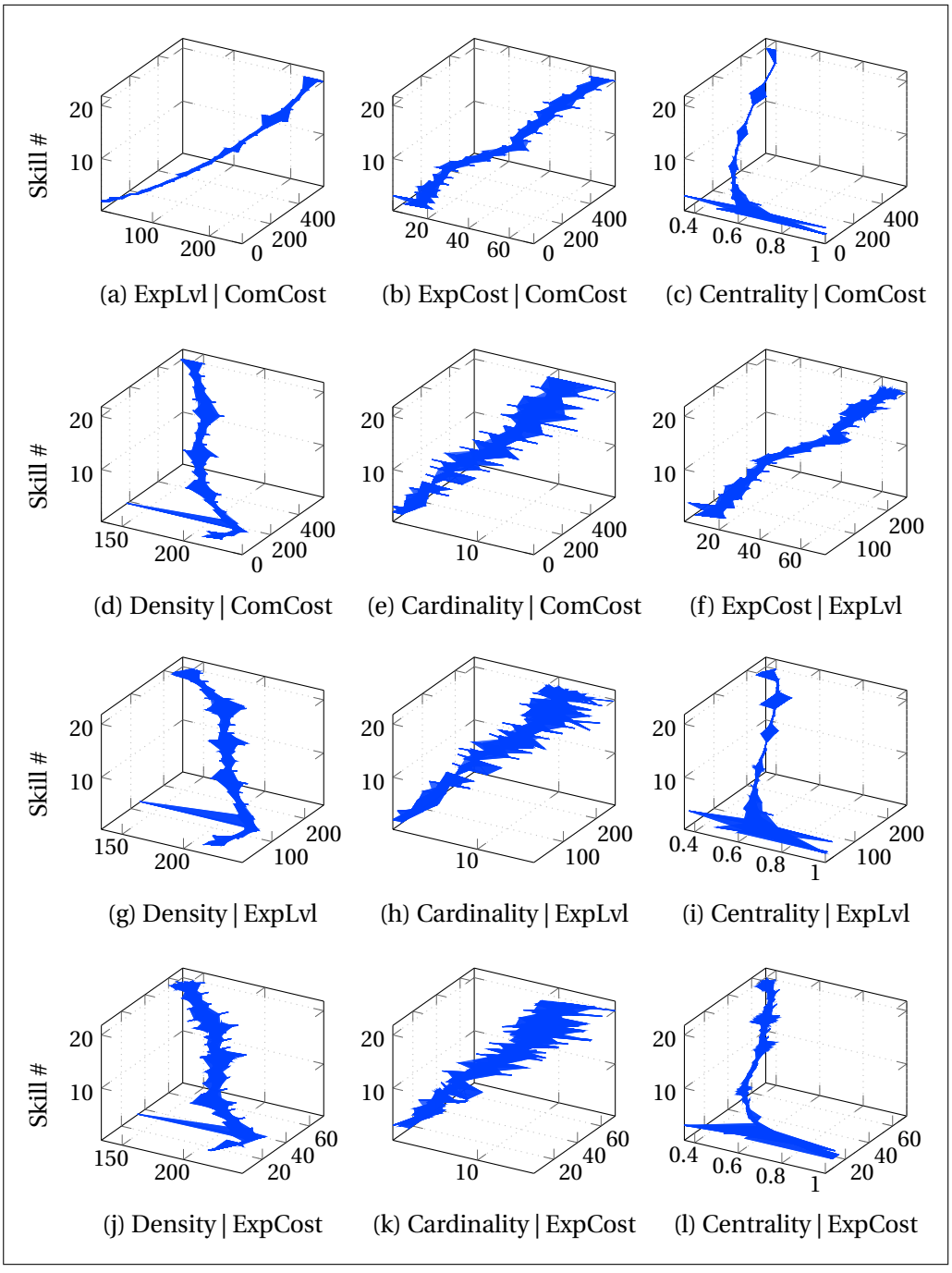


Figure 8.3: Sensitivity Analysis on the parameters and the metrics of ChemoTF given various required skills

Statement 8.5.2. ***(Effect of Communication Cost on Expert Cost)***

There is +77% correlation between Communication Cost and Expert Cost in ChemoTF which 12% depends on the number of skills.

In addition, the graphs (c) and (d) of Figure 8.3 demonstrate that the effect of $ComCost$ on the centrality and density is negative. Apart from small teams, the value for these metrics fall dramatically at the rate of 81% and 79% as $ComCost$ increases. The reason is that teams with higher $ComCost$ contain less communication channels or less edges between their experts compared to the teams with lower $ComCost$. Considering that the number of edges have a positive and direct effect on the centrality and density, both metrics decline for the teams accordingly. This is not a new observation and has already been fully discussed in [126].

Finally, $ComCost$ has +72% correlation with cardinality. This effect has been demonstrated on graph (e) of the Figure 8.3. The main reason is that the elevation of $ComCost$ causes the team to be more diverse and less connected. The growth of the number of experts is the compensation payed by ChemoTF to keep the communication effective. This effect is 63% dependant upon the skills and becomes even more dependant as the number of skills rises. This suggests that ChemoTF chooses wider range of experts for large tasks than for small tasks. There is simply more room for manoeuvre when the number of required skills are abundant.

8.5.2 Effect of Expertise Level

The effect of $ExpLvl$ on various metrics has been demonstrated in graphs (f), (g), (h), and (i) of the Figure 8.3. As discussed earlier, there is a positive correlation between $ExpLvl$ and $ExpCost$ depicted in graph (f) which corresponds to the correlation between $ComCost$ and $ExpCost$ demonstrated in graph (b). $ExpCost$ rises according to $ExpLvl$ because teams with higher $ExpLvl$ are more costly. It is common sense that individuals with higher expertise are competitively more costly than those with less

expertise. As ChemoTF forms teams with the highest level of expertise possible, $ExpLvl$ rises according to the size of the tasks. This rise causes an increase in $ExpCost$ consequently. This finding can be expressed as:

Statement 8.5.3. ***(Effect of Expertise Level on Expert Cost)***
There is +83% *correlation between Expertise Level and Expert Cost in ChemoTF which* 16% *depends on the number of required skills.*

Given the correlation between $ExpLvl$ and $ComCost$ as an effect independent of skills discussed earlier, the effect of $ExpLvl$ on the density, cardinality, and centrality is expected to be similar to the effect of $ComCost$ on these metrics. As illustrated in graphs (g), (h), and (i) of the Figure 8.3, the values of metrics rise and fall similarly for $ComCost$ and $ExpLvl$.

8.5.3 Effect of Expert Cost

The effect of $ExpCost$ on the metrics has been depicted on the graphs (j), (k), and (l) of the Figure 8.3. As mentioned earlier in Statements 8.5.3 and 8.5.2, both $ComCost$ and $ExpLvl$ have positive effects on $ExpCost$. Given these positive effects, the effect of $ExpCost$ on the density, cardinality and centrality is expected to be similar to the effect of $ComCost$ and $ExpLvl$ on these metrics. I calculated the effect within the range of 83% to 91% which is noticeable by comparing graph (j) to (g) and (d) for the density, graph (k) to (h) and (e) for the cardinality, and finally graph (l) to (i) and (c) for the centrality.

In addition, the effect of $ExpCost$ on the metrics is found 18% more dependant to skills compared to this effect for $ComCost$ and $ExpLvl$. This is because the effect of $ComCost$ and $ExpLvl$ on $ExpCost$ also slightly depends on the number of skills. This is why the graphs are slightly wider towards the z axis where the number of skills has been represented. In order to expand this observation for all the parameters, the statement bellow is given:

Statement 8.5.4. ***(Effect of ChemoTF Parameters on Metrics)***

The parameters Communication Cost, Expertise Level, and Expert Cost have similar effects on the metrics density, cardinality and centrality. This effect is positive on cardinality and negative on the density and centrality.

Analysing the experimental results of ChemoTF with respect to various metrics demonstrates that my algorithm forms teams in which members can communicate effectively with the rate of 100% of the required communication level for tasks consisting of up to 20 skills. This level of communication is particularly effective for tasks with 5 to 15 skills and reaches to the highest boundary of 90% of the required communication level for larger tasks. On the other hand, teams formed by ChemoTF possess the best ratio of the highest expertise level to the lowest communication cost at the rate of 1.04 for the tasks with 5 to 15 skills while remaining tractable for larger tasks. The cost of the team never exceeds the expected value and remains 100% tractable for any task.

The 98% goodness of fit for cost function and 99% goodness of fit for Chemistry Level, Communication Cost, and Expertise Level achieved in this experiment manifest the accuracy of the results. Evaluating the result of ChemoTF against three state-of-the-art algorithms suggests that my approach is better with respect to well-known metrics and at least 46% more predictable regardless of the size or complexity of the input.

Finally, sensitivity analysis suggests +79% correlation between Communication Cost and Expertise Level, +77% correlation between Communication Cost and Expert Cost, and +83% correlation between Expertise Level and Expert Cost in ChemoTF. The majority of the correlations are independent of the input. However the number of teams or the cardinality of the team is 63% dependant to the number of skills and becomes even more dependant as the number of required skills rise.

The next chapter concludes this thesis by looking back at the road taken to achieve the objectives and indicating the open roads for future research.

"With four parameters I can fit an elephant, and with five I can make him wiggle his trunk."

John von Neumann [47]

Chapter 9

Conclusions

Given a social network and a task consisting of required skills, Team Formation (TF) aims at forming a team of experts who cover the required skills and communicate effectively with one another. However prior studies consider effective communication as communication with minimum cost, disregarding other aspects of the quality of successful teams. As described in Hypothesis 1.1, assuming the success of a team rely upon various parameters, the minimum communication cost requirement can be relaxed to make room for other parameters such as the overall expertise of teams members. I also emphasised in Hypothesis 1.2 that given a quantifiable chemistry between skills, a threshold can be set according to the task reflecting the communication cost required to perform the task. Finally, I outlined in Hypothesis 1.3 that assuming costly experts lead to costly projects, teams with the least cost are likely to succeed.

Based on the hypotheses above, I proposed the Chemistry Oriented Team Formation (ChemoTF) and experimented it on a large corpus. I found that ChemoTF forms communicative, cost-effective, and highly expert teams. Evaluating ChemoTF results against state-of-the-art algorithms showed that team formed by ChemoTF have higher qualities with respect to well-known metrics. This chapter summarises my contributions towards TF in the field of computer science and outlines future work.

9.1 Contributions

TF has been often considered as the problem of finding teams with minimum cost of communication. This view over the problem is subjective and far from ideal. It is debatable whether teams with minimum overall communication cost can in fact communicate effectively. Given the complex multi-faceted relationship between multi-skilled experts, communication cost is expected to be dynamic and change according to the skills employed during an experience. Thus simplifying relationships into a single channel with a static cost value and forming teams according to the minimum cost overlooks the existing aspects of relationships. In addition, forming teams based on the lowest communication cost disregards teams with an overall higher expertise level which can still satisfy required communication level. This raises a profound question that this study has addressed:

- *How to form cost-effective team of experts with the highest level of expertise possible that can cover the required skills in a given task and its members are able to communicate with each other effectively?*

In addressing this question in this thesis, I proposed a novel approach towards team formation in social networks called Chemistry Oriented Team Formation (ChemoTF). ChemoTF defines effective communication as the expected level of communication required to perform a task and measures it using an innovative metric called Chemistry Level. Teams that can satisfy this measure of required communication are considered to communicate in an effective manner. Chemistry Level takes away the emphasis from finding teams with minimum communication cost and makes it possible to consider teams with the highest possible Expertise Level. In order to insure the selected experts are not unusually costly, ChemoTF takes advantage of another metric called Expert Cost and filters teams with higher personnel cost compared to the average cost of experts possessing the same skills throughout the social network.

The experimental results suggests that the algorithm I designed for ChemoTF forms teams in which members can communicate effectively with the rate of 100% of the required communication level for tasks consisting of up to 20 skills. This is particularly effective for tasks with 5 to 15 skills and reaches the highest boundary of 90% of the required communication level for larger tasks while remaining tractable for larger tasks. The cost of the team never exceeds the expected value and remains 100% tractable for any task.

More specifically, the contributions of this study are summarised by providing objective answers to the following research questions in the context of computer science:

1. *How to estimate dynamic communication cost which is computed according to the communication channel employed in a relationship?*

   I considered communication cost a dynamic variable which changes according to dimensions of multinomial relationships. Thanks to novel definition of areas of expertise serving as the dimensions of multidimensional social network, Dynamic Communication Cost is derived from relationships between experts according to the skills they have used during their mutual experiences.

2. *How to discover the required communication level for a given task according to the chemistry level between the skills it contains?*

   I introduced Chemistry Level as a required level of communication and measured it according the extent a pair of skills share the same areas of expertise within all the relationships in the network. The experimental results suggests that this measure is 100% effective and leads to better results compared to state-of-the-art algorithms. The 99% goodness of fit for Chemistry Level and Communication Cost functions accentuates the accuracy of my approach.

3. *How to quantify the expertise level of a team of experts?*

I quantified Expertise Level of an expert based on the number of experience he or she has had using a particular skill. This means that my estimation of Expertise Level is more accurate compared to similar approaches. This is verified by 99% goodness of fit for Expertise Level function. Using this estimation, my approach produced up to 73% better results compared to state-of-the-art algorithms.

4. *How to ensure the formed teams are cost-effective?*

   I employed a metric called Expert Cost to filter experts whose personnel cost is higher than the average cost of the experts possessing the sames skills throughout the social network. The experimental results suggests that my approach can form teams with the up to 45.8% lower Expert Cost with the success rate of 100% compared to state-of-the-art algorithms. The 98% goodness of fit for Expert Cost functions determines the accuracy of my implementation.

5. *How to design a mechanism in which all the essential parameters of team formation are incorporated?*

   I devised an approach called Chemistry-Oriented Team Formation (ChemoTF) in which cost-effective teams are formed based on the required level of communication and maximum level of expertise. Using this approach, I managed to produce results that are 46% more predictable and up to 73%, 11.5%, 48.34%, 23%, and 8.8% better with respect to well-known metrics including Expertise Level, Expert Cost, centrality, density, and cardinality compared to state-of-the-art algorithms.

In order to experiment ChemoTF, I compiled a corpus called Comprehensive Scholarly Corpus (CompScholarCorp) and used it a test-bed. This corpus features 472,365 experts, 24,500 skills, 1,044,454 experiences, and 3,466,85 citations for which 160 areas of expertise were generated using machine learning techniques. The experimental results suggests

that ChemoTF forms teams with the highest level of expertise and effective communication within the range of required communication level. Teams formed by ChemoTF for covering 5 to 15 skills were found to have low Communication Cost and high Expertise Level which is 53% better compared to smaller tasks and 76% better compared to larger tasks. The formed teams are 100% cost-efficient with respect to the expected cost regardless of the number of skills.

The result of ChemoTF were evaluated against three state-of-the-art TF approaches including RarestFirst, EnSteiner, and MinSD with respect to well-known metrics. It was found that ChemoTF consistently generates teams with higher qualities in a more predictable fashion. The performance of ChemoTF with respect to time was found to be ideal for covering up to 15 skills and tractable for larger tasks. In addition, the sensitivity analysis conducted on various metrics employed in evaluation suggested that these metrics are up to 84% independent of the input.

This thesis contributes to the concept of team formation in the field of computer science by proposing ChemoTF as a novel and unique approach. This approach overcomes the challenges encountered while finding team of experts that can perform tasks with the best possible outcome. To the best of my knowledge, the scale of the experiment conducted for ChemoTF is unprecedented throughout the literature and the results of this experiment was better than prior TF algorithms. This research brings to light the missing pieces of team formation in social networks and opens up new doors for future research.

Apart from the contributions made towards optimising team formation, the compiled corpus itself is a valuable output for future research. It includes unique information such as keywords and abstracts that corpora with the same nature and comparable scale has not been able to provide. Moreover, the design CompScholarCorp guarantees its compatibility to all prior and future DBLP-oriented approaches by which they can be repeated or expanded for further research.

9.2 Future Work

The approach discussed in this thesis provides a number of opportunities in the area of TF. One possible next step in this research would be include other parameters of a successful teams which were not covered in this study. These parameters include but not limited to: *personal commitment*, *productivity*, *growth potential*, *availability*, *project time-frame*, *personal capacity*, *initiation*, *innovation*, *diversity in the team*, *level of trust between team members*, etc. Employing each one of these parameters could contribute towards the success of the team.

Another major step forward would be to turn ChemoTF into a more dynamic form of team formation approach in which the concept of *cause and effect* would be considered. By simulating the result of a teamwork on a project, the quality of the team could be assessed. Based on this assessment, a parameter could be devised to represent the reputation of the experts or the teams and utilised while forming future teams. It would also be possible to generate a scheme in which this reputation would decay over time. Such a scheme would be more inline with reality.

Finally, ChemoTF can be implemented as an interesting feature for existing online social networks such as Facebook [1], Twitter [2], LinkedIn [3], ResearchGate [4], Academia [5], etc. Using the functionalities ChemoTF provides, project managers would be able to receive recommended teams based on the required skills for their projects. This capability can be further enhanced by devising parameters based on the information available in these social networks such as past experience, user-defined skills, geographical location, personal preferences, endorsements, friendship status, etc.

[1] www.facebook.com
[2] www.twitter.com
[3] www.linkedin.com
[4] www.researchgate.net
[5] www.academia.edu

Appendix A

Measuring the Divergence

The aim of Variational Inference is to find parameters such that the Variational distribution q is close to its posterior p. In order to be able to apply convergence methods to calculate the closeness of q and p, LDA assumes that Variational distribution is mean-field [127] as defined bellow:

$$q(\vec{\theta}, \vec{z}) = \prod_{e} q(\theta_e|\gamma_e) \prod_{n} q(z_{e,n}|\varphi_{e,n}) \tag{A.1}$$

In this equation, multinomial θ for each experience e is drawn from a Dirichlet distribution γ, and latent AoE assignments z is derived from a multinomial distribution φ. In addition, the components $\vec{\theta}$ and $\vec{z}$ are none-negative vectors of length K. $\vec{\theta}$ is the Variational experience distribution over γ_d whereas $\vec{z}$ is the Variational token distribution over $\varphi_{d,n}$.

One classic method which measures amount of information lost when q is used to approximate p is *Kullback-Leiber* (KL) divergence [85]. q and p are called close if the result of divergence is low which can only happen if q and p are both high. KL divergence is formulated bellow:

$$D_{KL}(q||p) \equiv \mathbb{E}_q[\log \frac{q(z)}{p(z|x)}] \tag{A.2}$$

Considering the definition of conditional probability of random variables z and x, KL divergence can be written as bellow:

$$D_{KL}(q||p) = -\overbrace{\mathbb{E}_q[\log p(z,x)]}^{\text{A}} - \overbrace{\mathbb{E}_q[\log q(z)]}^{\text{B}} + \overbrace{\log p(x)}^{\text{C}} \tag{A.3}$$

From *Jensen's Inequality* for concave functions [40], it is proved the average of the function (concave evaluated at two points) is less than the function applied to the average of the two points. Considering expectations, this inequality can be generalised to multiple points as bellow:

$$f(\mathbb{E}[x]) \geq \mathbb{E}[f(x)] \tag{A.4}$$

On the other hand, from the log probability of data [48] for latent variable z, the component C of the Equation A.2 can be rewritten as bellow:

$$\log p(x) = \log \int_z p(x,z)\,dz_k = \log \int_z \mathbb{E}_q[\log \frac{p(x,z)}{q(z)}]\,dz_k \tag{A.5}$$

Applying Jensen's inequality to the equation above, results in:

$$\overbrace{\log p(x)}^{\text{C}} \geq \overbrace{\mathbb{E}_q[\log p(z,x)]}^{\text{A}} - \overbrace{\mathbb{E}_q[\log q(z)]}^{\text{B}} \tag{A.6}$$

It is noticeable that this formula and Equation A.3 are constructed by the same terms. The term A refers to *Evidence Lower Bound* (ELBO) [21] and B is called *Entropy* [115]. Since the term C is independent of q, maximising ELBO has the same effect as minimising KL divergance. Expanding terms A and B, the general objective function for Variational inference can be devised as demonstrated in Equation A.7. This is the same objective function discussed in Chapter 6 for which Algorithm 4 was presented.

$$\begin{aligned}\mathscr{L}(\gamma,\varphi;\alpha,\beta) = &\overbrace{\mathbb{E}_q[\log p(\theta|\alpha)] + \mathbb{E}_q[\log p(z|\theta)] + \mathbb{E}_q[\log p(s|z,\beta)]}^{\text{ELBO}} \\ &\underbrace{-\mathbb{E}_q[\log q(\theta)] - \mathbb{E}_q[\log q(z)]}_{\text{ENTROPY}}\end{aligned} \tag{A.7}$$

The next appendix describes how to calculate each one of the expectations in ElBO and ENTROPY terms of this objective function.

Appendix B

Calculating Expectations

In order to calculate the expectations in ELBO given in Equation A.7, one needs to calculate the expectation of log Dirichlet first. Recall from Equation 6.1 that Dirichlet is parameterised by a vector α, and described by a multivariate continuous probability distribution Γ as bellow:

$$Dir(\alpha_1,...,\alpha_T) = \frac{\Gamma(\sum_j \alpha_j)}{\prod_j \Gamma(\alpha_j)} \prod_{j=1}^{T} p_j^{\alpha_j - 1} \tag{B.1}$$

Considering the definition of Dirichlet, the expectation of log Dirichlet can be calculated by the following equation where Ψ is the digamma function which is the first derivative of $\log\Gamma$:

$$\mathbb{E}_{dir}[\log p(\theta_i|\alpha)] = \Psi(\alpha_i) - \Psi(\sum_j \alpha_j) \tag{B.2}$$

Having the expectation of log Dirichlet in hand, the expectations of the Dirichlet distribution in ELBO are calculated as bellow:

$$\begin{cases} \mathbb{E}_q[\log p(\theta|\alpha)] = \log\Gamma(\sum_i \alpha_i) - \sum_i \log\Gamma(\alpha_i) + \sum_i (\alpha_i - 1)(\Psi(\gamma_i) - \Psi(\sum_j \gamma_j)) \\ \mathbb{E}_q[\log p(z|\theta)] = \sum_n \sum_i \varphi_{n,i}(\Psi(\gamma_i) - \Psi(\sum_j \gamma_j)) \\ \mathbb{E}_q[\log p(s|z,\beta)] = \sum_v^V \sum_i^K \varphi_{n,i} s_{e,n}^v \log \beta_{i,v} \end{cases} \tag{B.3}$$

The equations above calculate the expectations of ELBO of the objective function. Recall from Equation A.7 that the second part of the function reflects Entropy. In order to calculate the expectations in Entropy, first the entropy of Dirichlet is calculated using the equation bellow:

$$\mathbb{H}_q[\gamma] = -\log\gamma(\sum_j \gamma_j) + \sum_i \log\Gamma(\gamma_i) - \sum_i (\gamma_i - 1)(\Psi(\gamma_i) - \gamma(\sum_{j=1}^{K} \gamma_j)) \tag{B.4}$$

Having the entropy of Dirichlet in hand, the expectations in the Entropy term of the objective function is calculated by the entropy of multinomial as bellow:

$$\mathbb{H}_q[\varphi_{e,n}] = -\sum_i \varphi_{e,n,i} \log\varphi_{e,n,i} \tag{B.5}$$

Plugging the Equation B.3 and B.5 into the Equation A.7, the complete objective function for Variational inference can be written as follows:

$$\begin{aligned}\mathscr{L}(\gamma,\varphi;\alpha,\beta) &= \log\Gamma(\sum_{j=1}^{K}\alpha_j) - \sum_{i=1}^{K}\log\Gamma(\alpha_i) + \sum_{i=1}^{K}(\alpha_i - 1)(\Psi(\gamma_i) - \Psi(\sum_{j=1}^{K}\gamma_j)) \\ &+ \sum_{n=1}^{N}\sum_{i=1}^{K}\varphi_{n,i}(\Psi(\gamma_i) - \Psi(\sum_{j=1}^{K}\gamma_j)) + \sum_{n=1}^{N}\sum_{i=1}^{K}\sum_{j=1}^{V}\varphi_{n,i}s_n^j\log\beta_{i,j} - \log\Gamma(\sum_{j=1}^{K}\gamma_j) \\ &+ \sum_{i=1}^{K}\log\Gamma(\gamma_j) - \sum_{i=1}^{K}(\gamma_i - 1)(\Psi(\gamma_i) - \Psi(\sum_{j=1}^{K}\gamma_j)) - \sum_{n=1}^{N}\sum_{i=1}^{K}\varphi_{n,i}\log\varphi_{n,i}\end{aligned} \tag{B.6}$$

Bibliography

[1] Acharya, S., and Chellappan, S. Statistics. In *Pro Tableau: A Step-by-Step Guide*. Apress, (2017), pp. 495–545.

[2] Agrawal, R., Golshan, B., and Terzi, E. Grouping students in educational settings. In *Proceedings of ACM International Conference on Knowledge Discovery and Data Mining* (2014), pp. 1017–1026.

[3] Agustín-Blas, L. E., Salcedo-Sanz, S., Ortiz-García, E. G., and Portilla-Figueras, A. Team formation based on group technology: A hybrid grouping genetic algorithm approach. *Computers and Operations Research 38*, 2 (2011), pp. 484–495.

[4] Aho, A., and Ullman, V. *Principles of Computer Science Series*. C Edition. WH Freeman, (1994). pp. 15–16.

[5] An, A., Kargar, M., and ZiHayat, M. Finding affordable and collaborative teams from a network of experts. In *Proceedings of SIAM International Conference on Data Mining* (2013), pp. 587–595.

[6] Anagnostopoulos, A., Becchetti, L., Castillo, C., Gionis, A., and Leonardi, S. Power in unity: Forming teams in large-scale community systems. In *Proceedings of ACM Conference on Information and Knowledge Management* (2010), pp. 599–608.

[7] Anagnostopoulos, A., Becchetti, L., Castillo, C., Gionis, A., and Leonardi, S. Online team formation in social networks. In

Proceedings of ACM International Conference on World Wide Web (2012), pp. 839–848.

[8] Andrieu, C., de Freitas, N., Doucet, A., and Jordan, M. I. An introduction to mcmc for machine learning. *Machine Learning 50*, 1 (2003), pp. 5–43.

[9] Arkin, E. M., and Hassin, R. Minimum-diameter covering problems. *Networks 36*, 3 (2000), pp. 147–155.

[10] Awal, G. K., and Bharadwaj, K. K. Team formation in social networks based on collective intelligence: an evolutionary approach. *Applied Intelligence 41*, 2 (2014), pp. 627–648.

[11] Backstrom, L., Huttenlocher, D., Kleinberg, J., and Lan, X. Group formation in large social networks: Membership, growth, and evolution. In *Proceedings of ACM International Conference on Knowledge Discovery and Data Mining* (2006), pp. 44–54.

[12] Bai, Q., and Zhang, M. A flexible and reasonable mechanism for self-interested agent team forming. *Multiagent and Grid Systems 4*, 1 (2008), pp. 85–101.

[13] Barabási, A.-L., and Albert, R. Emergence of scaling in random networks. *Science 286*, 5439 (1999), pp. 509–512.

[14] Baroni, M., Bernardini, S., Ferraresi, A., and Zanchetta, E. The wacky wide web: a collection of very large linguistically processed web-crawled corpora. *Language Resources and Evaluation 43*, 3 (2009), pp. 209–226.

[15] Bayer, R., and McCreight, E. M. Organization and maintenance of large ordered indexes. *Acta Informatica 1*, 3 (1972), pp. 173–189.

[16] Baykasoglu, A., Dereli, T., and Das, S. Project team selection using fuzzy optimization approach. *Cybernetics and Systems 38*, 2 (2007), pp. 155–185.

[17] Beck, K. *Test-driven Development: By Example*. Kent Beck signature book. Addison-Wesley, (2003). pp. 20–55.

[18] Bhowmik, A., Borkar, V. S., Garg, D., and Pallan, M. Submodularity in team formation problem. In *Proceedings of SIAM International Conference on Data Mining* (2014), pp. 893–901.

[19] Binstock, C., Peterson, D., Smith, M., Wooding, M., Dix, C., and Galtenberg, C. *The XML Schema Complete Reference*. Addison-Wesley Longman Publishing, (2002). pp. 77-180.

[20] Bishop, C. *Pattern Recognition and Machine Learning*, 1 ed. Information Science and Statistics. Springer-Verlag, (2006). pp. 462–463.

[21] Blei, D. M., Jordan, M. I., et al. Variational inference for dirichlet process mixtures. *Bayesian analysis 1*, 1 (2006), pp. 121–143.

[22] Blei, D. M., Ng, A. Y., and Jordan, M. I. Latent dirichlet allocation. *The Journal of Machine Learning Research 3* (2003), pp. 993–1022.

[23] Blenko, M., Mankins, M., and Rogers, P. *Decide & Deliver: 5 Steps to Breakthrough Performance in Your Organization*. Harvard Business Review Press, (2010). pp. 87–88.

[24] Boehm, B. W., Gray, T. E., and Seewaldt, T. Prototyping vs. specifying: A multi-project experiment. In *Proceedings of IEEE International Conference on Software Engineering* (1984), pp. 473–484.

[25] Bonacich, P. Power and centrality: A family of measures. *American Journal of Sociology 92*, 5 (1987), pp. 1170–1182.

[26] Borgatti, S. P. Centrality and network flow. *Social Networks 27*, 1 (2005), pp. 55–71.

[27] Bozzon, A., Brambilla, M., Ceri, S., Silvestri, M., and Vesci, G. Choosing the right crowd: Expert finding in social networks. In *Proceedings of ACM International Conference on Extending Database Technology* (2013), pp. 637–648.

[28] Buccafurri, F., Lax, G., Nicolazzo, S., Nocera, A., and Ursino, D. Driving global team formation in social networks to obtain diversity. In *Web Engineering*, S. Casteleyn, G. Rossi, and M. Winckler, Eds., vol. 8541 of *Lecture Notes*. Springer, (2014), pp. 410–419.

[29] Bulka, B., Gaston, M., and desJardins, M. Local strategy learning in networked multi-agent team formation. *Autonomous Agents and Multi-Agent Systems 15*, 1 (2007), pp. 29–45.

[30] Burton, S. H., and Giraud-Carrier, C. G. Discovering social circles in directed graphs. *ACM Transactions on Knowledge Discovery from Data 8*, 4 (2014), pp. 1–27.

[31] Cao, C. C., She, J., Tong, Y., and Chen, L. Whom to ask?: Jury selection for decision making tasks on micro-blog services. *Proceedings of the VLDB Endowment 5*, 11 (2012), pp. 1495–1506.

[32] Cheatham, M., and Cleereman, K. Application of social network analysis to collaborative team formation. In *IEEE Symposium on Collaborative Technologies and Systems* (2006), pp. 306–311.

[33] Chen, S.-J., and Lin, L. Modeling team member characteristics for the formation of a multifunctional team in concurrent engineering. *IEEE Transactions on Engineering Management 51*, 2 (2004), pp. 111–124.

[34] Codd, E. F. A relational model of data for large shared data banks. *Communications of the ACM 13*, 6 (1970), pp. 377–387.

[35] Codd, E. F. Normalized data base structure: A brief tutorial. In *Proceedings of ACM Workshop on Data Description, Access and Control* (1971), pp. 1–17.

[36] Cohen, S. G., and Bailey, D. E. What makes teams work: Group effectiveness research from the shop floor to the executive suite. *Journal of Management 23*, 3 (1997), pp. 239–290.

[37] Coleman, T. F., and Moré, J. J. Estimation of sparse jacobian matrices and graph coloring problems. *Journal on Numerical Analysis 20*, 1 (1983), pp. 187–209.

[38] Connor, U., and Upton, T. *Applied Corpus Linguistics: A Multidimensional Perspective*. Language and computers : Studies in practical linguistics. Rodopi, (2004). pp. 194–195.

[39] Corgnet, B. Team formation and self-serving biases. *Journal of Economics & Management Strategy 19*, 1 (2010), pp. 117–135.

[40] Cvetkovski, Z. Convexity, jensen's inequality. In *Inequalities: Theorems, Techniques & Selected Problems*. Springer, (2012), pp. 69–77.

[41] Date, C. *An Introduction to Database Systems*. Systems programming series. Addison-Wesley, (1995). pp. 671-674.

[42] Dijkstra, E. W. Notes on structured programming, (1970). pp. 6–7.

[43] Dorn, C., and Dustdar, S. Composing near-optimal expert teams: A trade-off between skills and connectivity. In *On the Move to Meaningful Internet*, R. Meersman and T. Dillon, Eds., vol. 6426 of *Lecture Notes*. Springer, (2010), pp. 472–489.

[44] Dorn, C., Skopik, F., Schall, D., and Dustdar, S. Interaction mining and skill-dependent recommendations for multi-objective team composition. *Data And Knowledge Engineering 70*, 10 (2011), pp. 866–891.

[45] Dowling, M. J., Roering, W. D., Carlin, B. A., and Wisnieski, J. Multifaceted relationships under coopetition. *Journal of Management Inquiry 5*, 2 (1996), pp. 155–167.

[46] Dumais, S. T. Latent semantic analysis. *Annual Review of Information Science and Technology 38*, 1 (2004), pp. 188–230.

[47] Dyson, F. Turning points: A meeting with enrico fermi. *Nature 427* (2004), pp. 296–297.

[48] Edwards, A. W. F. *Likelihood*. Cambridge Press, (1972). pp. 5–85.

[49] Einstein, A. Infeld, L. The mechanical scaffold. In *The Evolution of Physics*. Simon & Schuster, (1961), pp. 158–159.

[50] Emerson, J. A., and Williams, D. M. The multifaceted relationship between physical activity and affect. *Social and Personality Psychology Compass 9*, 8 (2015), pp. 419–433.

[51] Farhadi, F., Sorkhi, M., Hashemi, S., and Hamzeh, A. An effective expert team formation in social networks based on skill grading. In *Proceedings of IEEE Data Mining Workshops* (2011), pp. 366–372.

[52] Feynman, R. P. *Six Easy Pieces: Essentials of Physics, Explained by Its Most Brilliant Teacher*. No. 1 in Books. Helix, (1995). pp. 101–102.

[53] Fitsilis, P., Gerogiannis, V., and Anthopoulos, L. Software project team selection based on enterprise social networks. In *Industrial Engineering, Management Science*, M. Gen and K. J. Kim, Eds., vol. 349 of *Lecture Notes*. Springer, (2015), pp. 375–384.

[54] Fitzpatrick, E. L., and Askin, R. G. Forming effective worker teams with multi-functional skill requirements. *Computers and Industrial Engineering 48*, 3 (2005), pp. 593–608.

[55] Freeman, L. C. Centrality in social networks conceptual clarification. *Social Networks 1*, 3 (1978), pp. 215–239.

[56] Fruchterman, T. M. J., and Reingold, E. M. Graph drawing by force-directed placement. *Software: Practice and Experience 21*, 11 (1991), pp. 1129–1164.

[57] Gajewar, A., and Sarma, A. D. Multi-skill collaborative teams based on densest subgraphs. In *Proceedings of SIAM International Conference on Data Mining* (2012), pp. 165–176.

[58] Garg, N., Konjevod, G., and Ravi, R. A polylogarithmic approximation algorithm for the group steiner tree problem. *Journal of Algorithms 37*, 1 (2000), pp. 66–84.

[59] Gaston, M. E., and Desjardins, M. Social network structures and their impact on multi-agent system dynamics. In *Proceedings of the AAAI International Artificial Intelligence Research Society Conference* (2005), pp. 32–37.

[60] Gaston, M. E., Desjardins, M., and Simmons, J. Adapting network structures for efficient team formation. In *Proceedings of AAMAS Workshop on Learning and Evolution in Agent-based Systems* (2004), pp. 230–237.

[61] Geman, S., and Geman, D. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis Machine Intelligence PAMI*, 6 (1984), pp. 721–741.

[62] Girolami, M., and Kabán, A. On an equivalence between plsi and lda. In *Proceedings of ACM International Conference on Research and Development in Information Retrieval* (2003), pp. 433–434.

[63] Goldman, R., Shivakumar, N., Venkatasubramanian, S., and Garcia-Molina, H. Proximity search in databases. In *Proceedings of the Conference on Very Large Databases* (1998), pp. 26–37.

[64] Golshan, B., Lappas, T., and Terzi, E. Profit-maximizing cluster hires. In *Proceedings of ACM International Conference on Knowledge Discovery and Data Mining* (2014), pp. 1196–1205.

[65] Grady, J. *System Requirements Analysis*. McGraw-Hill Education, (1993). pp. 23–31.

[66] Griffiths, T., and Steyvers, M. A probabilistic approach to semantic representation. In *Proceedings of the Conference of the Cognitive Science Society* (2002), pp. 381–386.

[67] Halperin, E., and Krauthgamer, R. Polylogarithmic inapproximability. In *Proceedings of Annual ACM Symposium on Theory of Computing* (2003), pp. 585–594.

[68] Hirsch, J. E. An index to quantify an individual's scientific research output. *Proceedings of the National Academy of Sciences of the USA* (2005), pp. 16569–16572.

[69] Hodges, A. *Alan Turing: The Enigma*. Vintage, (1992). pp. 27–28.

[70] Horwitz, S. K., and Horwitz, I. B. The effects of team diversity on team outcomes: A meta-analytic review of team demography. *Journal of Management 33*, 6 (2007), 987–1015.

[71] Howe, J. The rise of crowdsourcing. *Wired 14*, 6 (2006), pp. 1–4.

[72] Jardim-Gonçalves, R., Müller, J., Mertins, K., and Zelm, M. *Enterprise Interoperability II: New Challenges and Approaches*. Springer, (2007). pp. 161–168.

[73] Johansson, S. Some aspects of the development of corpus linguistics in the 1970s and 1980s. In *Corpus Linguistics, An International Handbook*, L. A. and K. M., Eds. De Gruyter, (2008), pp. 33–53.

[74] Jones, C. *Applied Software Measurement: Assuring Productivity and Quality*. Computing. McGraw-Hill, (1997). pp. 157–188.

[75] Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., and Saul, L. K. An introduction to variational methods for graphical models. *Machine Learning 37*, 2 (1999), pp 183–233.

[76] Kargar, M., and An, A. Discovering top-k teams of experts with-/without a leader in social networks. In *Proceedings of ACM International Conference on Information and Knowledge Management* (2011), pp. 985–994.

[77] Kargar, M., and An, A. Teamexp: Top-k team formation in social networks. In *Proceedings of IEEE International Conference on Data Mining Workshops* (2011), pp. 1231–1234.

[78] Kargar, M., An, A., and Zihayat, M. Efficient bi-objective team formation in social networks. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases* (2012), pp. 483–498.

[79] Karp, R. M. Reducibility among combinatorial problems. In *Complexity of Computer Computations: Proceedings of a symposium on the Complexity of Computer Computations*, R. E. Miller and J. W. Thatcher, Eds., vol. 1. Springer, (1972), pp. 85–103.

[80] Kennedy, G. D. *An introduction to corpus linguistics*. Longman, (1998). pp. 185–195.

[81] Kilgarriff, A., and Grefenstette, G. Introduction to the special issue on the web as corpus. *Computational Linguistics 29*, 3 (2003), pp. 333–347.

[82] Klug, M., and Bagrow, J. P. Understanding the group dynamics and success. *Royal Society Open Science 3*, 4 (2016), pp. 1–11.

[83] KOTONYA, G., AND SOMMERVILLE, I. *Requirements engineering: processes and techniques*. Worldwide series in computer science. Wiley, (1998). pp. 35–40.

[84] KOZLOWSKI, S. W., AND ILGEN, D. R. Enhancing the effectiveness of work groups and teams. *Psychological Science in the Public Interest 7*, 3 (2006), pp. 77–124.

[85] KULLBACK, S., AND LEIBLER, R. A. On information and sufficiency. *The Annals of Mathematical Statistics 22*, 1 (1951), pp. 79–86.

[86] LAPPAS, T., LIU, K., AND TERZI, E. Finding a team of experts in social networks. In *Proceedings of ACM International Conference on Knowledge Discovery and Data Mining* (2009), pp. 467–476.

[87] LAPPAS, T., LIU, K., AND TERZI, E. A survey of algorithms and systems for expert location in social networks. In *Social Network Data Analytics*, C. C. Aggarwal, Ed. Springer, (2011), pp. 215–241.

[88] LI, C.-T., AND SHAN, M.-K. Composing activity groups in social networks. In *Proceedings of ACM International Conference on Information and Knowledge Management* (2012), pp. 2375–2378.

[89] LI, C.-T., SHAN, M.-K., AND LIN, S.-D. Context-based people search in labeled social networks. In *Proceedings of ACM Conference on Information and Knowledge Management* (2011), pp. 1607–1612.

[90] LI, C.-T., SHAN, M.-K., AND LIN, S.-D. On team formation with expertise query in collaborative social networks. *Knowledge and Information Systems 42*, 2 (2015), pp. 441–463.

[91] LI, K., LU, W., BHAGAT, S., LAKSHMANAN, L. V., AND YU, C. On social event organization. In *Proceedings of ACM Conference on Knowledge Discovery and Data Mining* (2014), pp. 1206–1215.

[92] Li, L., Tong, H., Cao, N., Ehrlich, K., Lin, Y.-R., and Buchler, N. Replacing the irreplaceable: Fast algorithms for team member recommendation. In *Proceedings of the 24th International Conference on World Wide Web* (2015), pp. 636–646.

[93] Liang, S., and de Rijke, M. Finding knowledgeable groups in enterprise corpora. In *Proceedings of ACM Conference on Research and Development in Information Retrieval* (2013), pp. 1005–1008.

[94] Lima-Mendez, G., and van Helden, J. The powerful law of the power law and other myths in network biology. *Molecular BioSystems 5* (2009), pp. 1482–1493.

[95] Lüdeling, A., Evert, S., and Baroni, M. Using web data for linguistic purposes. In *Language and Computers*, N. N. . C. B. Marianne Hundt, Ed., vol. 59 of *Corpus Linguistics and the Web*. Rodopi, (2007), pp. 7–24.

[96] Maghami, M., and Sukthankar, G. An agent-based simulation for investigating the impact of stereotypes on task-oriented group formation. In *Proceedings of the Conference on Social Computing, Behavioral-cultural Modeling and Prediction* (2011), pp. 252–259.

[97] Majumder, A., Datta, S., and Naidu, K. Capacitated team formation problem on social networks. In *Proceedings of ACM Conference on Knowledge Discovery and Data Mining* (2012), pp. 1005–1013.

[98] Martin, R. *Agile Software Development: Principles, Patterns, and Practices*. Alan Apt series. Pearson Education, (2003). pp. 43–66.

[99] Mayer-Schonberger, V., and Cukier, K. *Big Data: A Revolution That Will Transform How We Live, Work and Think*. Hodder & Stoughton, (2013). pp. 11–12.

[100] MILLER, G. A. The magical number seven, plus or minus two: some limits on our capacity for processing information. *Psychological review 63*, 2 (1956), pp. 81–97.

[101] MINKA, T., AND LAFFERTY, J. Expectation-propagation for the generative aspect model. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence* (2002), pp. 352–359.

[102] NAJAFLOU, Y., JEDARI, B., XIA, F., YANG, L. T., AND OBAIDAT, M. S. Safety challenges and solutions in mobile social networks. *IEEE Systems Journal 9*, 3 (2015), pp. 834–854.

[103] NESHATI, M., HASHEMI, S. H., AND BEIGY, H. Expertise finding in bibliographic network: Topic dominance learning approach. *IEEE Transactions on Cybernetics 44*, 12 (2014), pp. 2646–2657.

[104] NEWMAN, D. J. A test harness for maintaining unfamiliar software. In *Proceedings of the Conference on Software Maintenance* (1988), pp. 409–416.

[105] OWENSA, D. A., MANNIX, E. A., AND NEALE, M. A. Strategic formation of groups: Issues in task performance and team member selection. In *Composition*, D. H. Gruenfeld, Ed., vol. 1 of *Research on managing groups and teams*. Elsevier Science, (1998), pp. 149–165.

[106] PAGE-JONES, M. *The Practical Guide to Structured Systems Design: 2nd Edition*. Yourdon Press, (1988). pp. 10-15.

[107] PARAMESWARAN, A., VENETIS, P., AND GARCIA-MOLINA, H. Recommendation systems with complex constraints: A course recommendation perspective. *ACM Transactions on Information Systems 29*, 4 (2011), pp. 1–33.

[108] PERCIVAL, H. *Test-Driven Development with Python: Obey the Testing Goat*. Using Django, Selenium, and JavaScript. O'Reilly, (2014). pp. 120–121.

[109] Rangapuram, S. S., Bühler, T., and Hein, M. Towards realistic team formation in social networks based on densest subgraphs. In *Proceedings of ACM International Conference on World Wide Web* (2013), pp. 1077–1088.

[110] Renouf, A. from super-corpus to cyber-corpus. In *Corpus development 25 years on*, R. Facchinetti, Ed., vol. 62 of *Language and Computers*. Rodopi, (2007), pp. 27–49.

[111] Ricci, F., Rokach, L., and Shapira, B. Introduction to recommender systems handbook. In *Recommender Systems Handbook*, P. B. Kantor, Ed. Springer, (2011), pp. 1–35.

[112] Saltelli, A. Sensitivity analysis for importance assessment. *Risk Analysis 22*, 3 (2002), pp. 579–590.

[113] Saltelli, A., Ratto, M., Andres, T., Campolongo, F., Cariboni, J., Gatelli, D., and Saisana, M. Introduction to sensitivity analysis. In *Global Sensitivity Analysis. The Primer*. Wiley, (2008), pp. 1–51.

[114] Schwartz, M. *Principles of Electrodynamics*. Books on Physics. Dover, (2012). pp. 1–2.

[115] Shannon, C. E. A mathematical theory of communication. *The Bell System Technical Journal 27*, 3 (1948), pp. 379–423.

[116] Sharifi, S., and Pawar, K. S. Virtually co-located product design teams: Sharing teaming experiences after the event? *Operations & Production Management 22*, 6 (2002), pp. 656–679.

[117] Sharoff, S., Rapp, R., Zweigenbaum, P., and Fung, P. *Building and Using Comparable Corpora*. Theory and applications of natural language processing. Springer, (2013). pp. 1–17.

[118] Sinclair, J. *Corpus, Concordance, Collocation*. Oxford University Press, (1991). pp. 14–15.

[119] Sozio, M., and Gionis, A. The community-search problem and how to plan a successful cocktail party. In *Proceedings of ACM International Conference on Knowledge Discovery and Data Mining* (2010), pp. 939–948.

[120] Stallings, J., Vance, E., Yang, J., Vannier, M. W., Liang, J., Pang, L., Dai, L., Ye, I., and Wang, G. Determining scientific impact using a collaboration index. *Proceedings of the National Academy of Sciences of the USA 110*, 24 (2013), pp. 9680–9685.

[121] Steyvers, M., and Griffiths, T. Probabilistic topic models. *Handbook of latent semantic analysis 427*, 7 (2007), pp. 424–440.

[122] Tang, J., Zhang, J., Yao, L., Li, J., Zhang, L., and Su, Z. Arnetminer: Extraction and mining of academic social networks. In *Proceedings of ACM International Conference on Knowledge Discovery and Data Mining* (2008), pp. 990–998.

[123] Taylor, J. B. Building an interdisciplinary team. In *Perspectives on technology assessment*, S. Arnstein and A. Christakis, Eds., vol. 1. Jerusalem: Science and Technology Publisher, (1975), pp. 45–60.

[124] Teng, Y.-C., and Wang, J.-Z. Team formation with the communication load constraint in social networks. In *Trends and Applications in Knowledge Discovery and Data Mining*, W.-C. Peng and Wang, Eds., vol. 8643 of *Lecture Notes*. Springer, (2014), pp. 125–136.

[125] Tong, H., He, J., Wen, Z., Konuru, R., and Lin, C.-Y. Diversified ranking on large graphs: An optimization viewpoint. In *Proceedings of ACM International Conference on Knowledge Discovery and Data Mining* (2011), pp. 1028–1036.

[126] Valente TW, Coronges K, L. C. C. E. How correlated are network centrality measures? *Connections 1*, 28 (2008), pp. 16–26.

[127] WAINWRIGHT, M. J., AND JORDAN, M. I. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning 1*, 1 (2008), pp. 1–305.

[128] WALLACH, H. M., MURRAY, I., SALAKHUTDINOV, R., AND MIMNO, D. Evaluation methods for topic models. In *Proceedings of ACM International Conference on Machine Learning* (2009), pp. 1105–1112.

[129] WANG, X., AND ZHAO, Z. A comparative study of team formation in social networks. In *Database Systems for Advanced Applications*, M. Renz and Shahabi, Eds., vol. 9049 of *Lecture Notes*. Springer, (2015), pp. 389–404.

[130] WANG, X., ZHAO, Z., AND NG, W. Ustf: A unified system of team formation. *IEEE Transactions on Big Data 2*, 1 (2016), pp. 70–84.

[131] WERBEL, J. D., AND JOHNSON, D. J. The use of person–group fit for employment selection: A missing link in person–environment fit. *Human Resource Management 40*, 3 (2001), pp. 227–240.

[132] WI, H., OH, S., MUN, J., AND JUNG, M. A team formation model based on knowledge and collaboration. *Expert Systems with Applications: An International Journal 36*, 5 (2009), pp. 9121–9134.

[133] WILLIAMS, K. Y., AND O'REILLY, C. A. Demography and diversity in organizations: A review of 40 years of research. *Research in Organizational Behavior 20* (1998), pp .77–140.

[134] WU, T., CHEN, L., HUI, P., ZHANG, C. J., AND LI, W. Hear the whole story: Towards the diversity of opinion in crowdsourcing markets. *Proceedings of the VLDB Endowment 8*, 5 (2015), pp. 485–496.

[135] XIE, M., LAKSHMANAN, L. V., AND WOOD, P. T. Breaking out of the box of recommendations: From items to packages. In *Proceedings of ACM Conference on Recommender Systems* (2010), pp. 151–158.

[136] Xu, Y., Ma, J., and Guo, X. Modeling researchers' characteristics for the formation of research team. In *Proceedings of Pacific Asia Conference on Information Systems* (2012), pp. 252–259.

[137] Yin, H., and Cui, B. Finding a wise group of experts in social networks. In *Advanced Data Mining and Applications*, J. Tang and King, Eds., vol. 7120 of *Lecture Notes*. Springer, (2011), pp. 381–394.

[138] Zhao, Q., Tian, Y., He, Q., Oliver, N., Jin, R., and Lee, W.-C. Communication motifs: A tool to characterize social communications. In *Proceedings of ACM International Conference on Information and Knowledge Management* (2010), pp. 1645–1648.

[139] Zhao, Z., Cheng, J., Wei, F., Zhou, M., Ng, W., and Wu, Y. Socialtransfer: Transferring social knowledge for cold-start cowdsourcing. In *Proceedings of ACM International Conference on Conference on Information and Knowledge Management* (2014), pp. 779–788.

[140] Zhao, Z., Ng, W., and Zhang, Z. Crowdseed: Query processing on microblogs. In *Proceedings of ACM International Conference on Extending Database Technology* (2013), pp. 729–732.

[141] Zhao, Z., Yan, D., Ng, W., and Gao, S. A transfer learning based framework of crowd-selection on twitter. In *Proceedings of ACM International Conference on Knowledge Discovery and Data Mining* (2013), pp. 1514–1517.

[142] Zheng, Y., Cheng, R., Maniu, S., and Mo, L. On optimality of jury selection in crowdsourcing. In *Proceedings of the International Conference on Extending Database Technology* (2015), pp. 193–204.

[143] Zhou, W., Jin, H., and Liu, Y. Community discovery and profiling with social messages. In *Proceedings of ACM International Conference on Knowledge Discovery and Data Mining* (2012), pp. 388–396.

[144] ZHU, H., CHEN, E., XIONG, H., CAO, H., AND TIAN, J. Ranking user authority with relevant knowledge categories for expert finding. *World Wide Web 17*, 5 (2014), pp. 1081–1107.

[145] ZHU, H., ZHOU, M., AND SEGUIN, P. Supporting software development with roles. *IEEE Transactions on Systems, Man, and Cybernetics 36*, 6 (2006), pp. 1110–1123.

[146] ZZKARIAN, A., AND KUSIAK, A. Forming teams: an analytical approach. *IIE Transactions 31*, 1 (1999), pp. 85–97.