5-21-2020

Investigation of Team Formation in Dynamic Social Networks

Kalyani Selvarajah
University of Windsor

Investigation of Team Formation in Dynamic Social Networks

by

Kalyani Selvarajah

A Dissertation
Submitted to the Faculty of Graduate Studies
through the School of Computer Science
in Partial Fulfillment of the Requirements for
the **Degree of Doctor of Philosophy**
at the University of Windsor

Windsor, Ontario, Canada
2020

Investigation of Team Formation in Dynamic Social Networks

by

Kalyani Selvarajah

APPROVED BY:

H. Usefi, External Examiner
Memorial University of Newfoundland

K. Pfaff
Faculty of Nursing

S. Samet
School of Computer Science

S. Saad
School of Computer Science

Z. Kobti, Co-Advisor
School of Computer Science

M. Kargar, Co-Advisor
School of Computer Science

March 18, 2020

Declaration of Co-Authorship/Previous Publication

I Co-Authorship

I hereby declare that this dissertation incorporates material that is the result of research conducted under the supervision of Dr. Ziad Kobti (Advisor) and Dr. Mehdi Kargar (Co-Advisor). The collaboration is covered in Chapters 2, 3, 4, 5, 6, 7 and 8 of the dissertation. In all cases, the key ideas, primary contributions, experimental designs, data analysis and interpretation, were performed by Kalyani Selvarajah (the candidate), Dr. Ziad Kobti and Dr. Mehdi Kargar (the supervisors) as primary authors and contributors.

In chapter 2, 3, 4 and 6, this dissertation incorporates the outcome of a joint research undertaken in collaboration with Dr. Pooya Moradian Zadeh under the supervision of Dr. Ziad Kobti and Dr. Mehdi Kargar.

Chapter 3 and 4 was co-authored with Dr. Kathryn Pfaff who supported in finalizing ideas and follow-up discussions.

In "Link Prediction by Analyzing Common Neighbors Based Subgraphs using Convolutional Neural Network", ECAI 2020, Mr. Kumaran Ragunathan has implemented the method. Kalyani Selvarajah participated in finalizing ideas and writing research paper under the supervision of Dr. Ziad Kobti.

In "Productive and Profitable Cluster Hire", FLAIRS 2019, Parth Patel has implemented major algorithms. Kalyani Selvarajah has implemented suportive algorithms and participated in finalizing ideas and writing research paper under the supervision of Dr. Ziad Kobti and Dr. Mehdi Kargar.

In chapter 3, Selvarajah Kalyani, Dr. Pooya Moradian Zadeh and Mr. Mohd Tazim Ishraque participated in writing the text of paper.

Chapter 8 was co-authored with Mr.Kumaran Ragunathan who participated in finalizing ideas and follow-up discussions.

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

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

II Declaration of Previous Publication

This dissertation includes the extended or original version of the papers that have been previously published/submitted for publication in peer reviewed conferences and journals, as follows:

Section	Full Citation	Status
Chapter 2	Selvarajah Kalyani, Pooya Moradian Zadeh, Mehdi Kargar, and Ziad Kobti. "A knowledge-based computational algorithm for discovering a team of experts in social networks." (KDD-DDD Workshop 2017).	Accepted
Chapter 2	Selvarajah Kalyani, Pooya Moradian Zadeh, Mehdi Kargar, and Ziad Kobti. "Identifying a Team of Experts in Social Networks using a Cultural Algorithm." The 10th International Conference on Ambient Systems, Networks and Technologies (ANT 2019): 477-484.	Published
Chapter 3	Selvarajah Kalyani, Pooya Moradian Zadeh, Ziad Kobti, Mehdi Kargar, Mohd Tazim Ishraque, and Kathryn Pfaff. "Team formation in community-based palliative care." In 2018 Innovations in Intelligent Systems and Applications (INISTA 2018), pp. 1-7. IEEE, 2018.	Published

Chapter 4	Selvarajah Kalyani, Pooya Moradian Zadeh, Ziad Kobti, Kathryn Pfaff, and Mehdi Kargar. "A Palliative Care Simulator and Visualization Framework." and Healthcare Systems, and Multimedia (KES-InMed 2019): 317.	Published
Chapter 5	Selvarajah Kalyani, Ziad Kobti, Mehdi Kargar. "Cultural Algorithms for Cluster Hires in Social Networks", The 11th International Conference on Ambient Systems, Networks and Technologies (ANT 2020).	Accepted
	Patel Parth, Kalyani Selvarajah, Ziad Kobti, and Mehdi Kargar. "Productive and Profitable Cluster Hire." In The Thirty-Second International Flairs Conference. FLAIRS 2019.	Published
Chapter 6	Selvarajah Kalyani, Ziad Kobti, Mehdi Kargar, Pooya Moradian Zadeh. "A Unified Framework for Effective Team Formation in Social Networks", the expert systems with application, an international journal.	Submitted
Chapter 7	Selvarajah Kalyani, Ziad Kobti, Mehdi Kargar. "Link Prediction by Analyzing Temporal Behavior of Vertices", The International Conference on Computational Science (ICCS 2020)	Accepted
	Kumaran Ragunathan, Kalyani Selvarajah, Ziad Kobti. "Link Prediction by Analyzing Common Neighbors Based Subgraphs using Convolutional Neural Network", the 24th European Conference on Artificial Intelligence (ECAI 2020)	Accepted
Chapter 8	Kalyani Selvarajah, Kumaran Ragunathan, Ziad Kobti, Mehdi Kargar. "Dynamic Network Link Prediction by Learning Effective Subgraphs using CNN-LSTM", The 2020 International Joint Conference on Neural Networks (IJCNN 2020)	Accepted

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

III General

I declare that, to the best of my knowledge, my dissertation does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my dissertation, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained a written permission from the copyright owner(s) to include such material(s) in my dissertation. I declare that this is a true copy of my dissertation, including any final revisions, as approved by my dissertation committee and the Graduate Studies office, and that this dissertation has not been submitted for a higher degree to any other University or Institution.

Abstract

Team Formation Problem (TFP) in Social Networks (SN) is to collect the group of individuals who match the requirements of given tasks under some constraints. It has several applications, including academic collaborations, healthcare, and human resource management. These types of problems are highly challenging because each individual has his or her own demands and objectives that might conflict with team objectives. The major contribution of this dissertation is to model a computational framework to discover teams of experts in various applications and predict the potential for collaboration in the future from a given SN. Inspired by an evolutionary search technique using a higher-order cultural evolution, a framework is proposed using Knowledge-Based Cultural Algorithms to identify teams from co-authorship and industrial settings. This model reduces the search domain while guiding the search direction by extracting situational knowledge and updating it in each evolution.

Motivated from the above results, this research examines the palliative care multidisciplinary networks to identify and measure the performance of the optimal team of care providers in a highly dynamic and unbalanced SN of volunteer, community, and professional caregivers. Thereafter, a visualization framework is designed to explore and monitor the evolution in the structure of the care networks. It helps to identify isolated patients, imbalanced resource allocation, and uneven service distribution in the network. This contribution is recognized by Hospice and the Windsor Essex Compassion Care Community in partnership with the Faculty of Nursing.

In each setting, several cost functions are attempted to measure the performance of the teams. To support this study, the temporal nature of two important evaluation metrics is analyzed in Dynamic Social Networks (DSN): dynamic communication cost and dynamic expertise level. Afterward, a novel generic framework for TFP is designed by incorporating essential cost functions, including the above dynamic cost functions. The Multi-Objective Cultural Algorithms (MOCA) is used for this pur-

pose. In each generation, it keeps track of the best solutions and enhances exploration by driving mutation direction towards unexplored areas. The experimental results reach closest to the exact algorithm and outperform well-known searching methods.

Subsequently, this research focuses on predicting suitable members for the teams in the future, which is typically a real-time application of Link Prediction. Learning temporal behavior of each vertex in a given DSN can be used to decide the future connections of the individual with the teams. A probability function is introduced based on the activeness of the individual. To quantify the activeness score, this study examines each vertex as to how actively it interacts with new and existing vertices in DSN. It incorporates two more objective functions: the weighted shortest distance and the weighted common neighbor index. Because it is technically a classification problem, deep learning methods have been observed as the most effective solution. The model is trained and tested with Multilayer Perceptron. The AUC achieves above 93%. Besides this, analyzing common neighbors with any two vertices, which are expected to connect, have a high impact on predicting the links. A new method is introduced that extracts subgraph of common neighbors and examines features of each vertex in the subgraph to predict the future links. The sequence of subgraphs' adjacency matrices of DSN can be ordered temporally and treated as a video. It is tested with Convolutional Neural Networks and Long Short Term Memory Networks for the prediction. The obtained results are compared against heuristic and state-of-the-art methods, where the results reach above 96% of AUC.

In conclusion, the knowledge-based evolutionary approach performs well in searching through SN and recommending effective teams of experts to complete given tasks successfully in terms of time and accuracy. However, it does not support the prediction problem. Deep learning methods, however, perform well in predicting the future collaboration of the teams.

Dedication

I would like to dedicate this dissertation to my mother, Anushihadevi and to my brother Jeyanthan for their endless love, support, and encouragement.

Acknowledgements

Firstly, I am deeply thankful to my advisors Dr. Ziad Kobti and Dr. Mehdi Kargar, for their continuous support of my PhD studies and related research for their extensive professional guidance, and for their immense knowledge. Besides academic and professional work, Dr. Ziad Kobti is an amazing professor of the School of Computer Science and mentored and appreciated me on my every little progress. Although Dr. Mehdi Kargar moved to Ryerson University, he guided me through Skype to give feedback and to listen to my problems with patient. Without their guidance and persistent help, this dissertation would not have been possible.

I am greatly indebted to my internal readers, Dr. Saeed Samet and Dr. Sherif Saad Ahmed for participating in my seminars, for taking time to read my dissertation, and for providing me with their valuable comments. I would like to thank my external reader and collaborator, Dr. Kathryn Pfaff who spent time with us in brainstorming the research on palliative care jointly with Hospice and the Windsor Essex Compassion Care Community.

I would also like to thank Dr. Pooya Moradian Zadeh for being part of my research work and supported me during my Ph.D. journey.

In addition, my thanks goes to International Business Machines (IBM) for providing us a high performance IBM Power System S822LC Linux Server, which supported to conduct my research experiments.

I owe special thanks to the staffs of School of Computer Science, Ms. Gloria Mensah, Ms. Melissa Robinet, Ms. Karen Bourdeau, Ms. Christine Weisener, Ms. Margaret Garabon, Ms. Tina Palmer, Mr. Maunzer Batal, Mr. Sanjay Chitte and Mr. Robert Mavrinac for their great support.

I would like to thank my mother and brother for their support in every success of my life. I am also very thankful to my close friends Kumaran Ragunathan, Ambikai Gajan and her family and all my lab mates for their moral support and listening to

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

Kalyani Selvarajah

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Social Network Analysis

Social networks are not only modern online varieties such as Facebook[1], Twitter[2], Instagram[3]. These can also be the kind of social structure that human beings have been assembling for a variety of reasons or the way how the proteins link to each other. Because the pattern of social structure is treated as a network, it grabbed the name Social Networks. It has so many hidden and powerful information which makes us see the world in an entirely new way. Social networks are intricate things of beauty and do not follow either a random or regular pattern. Social Networks Analysis (SNA) can be simply defined as an in-depth analysis of social network structure, dynamic nature, multi-relational aspects, the pattern of a relationship with social actors, and the available data along with them [1]. Since social networks use to represent the large-scale social structures and the underlying network structure in the real world is dynamic and large in size, called as complex system [2]. Moreover, these are ubiquitous and can be created from various fields such as co-author network,

[1]https://www.facebook.com/
[2]https://twitter.com/
[3]https://www.instagram.com/

Twitter friendship, LinkedIn profile[4], and protein to protein network. Therefore, the demand for contribution to the fundamental research of complex SNA has become high. Investigating complex social networks is extremely challenging. The study of complex social networks is limited in the literature because of unexpected challenges owing to the real-data. For example, the healthcare network analysis requires patient information. But for the privacy purpose, no one release real data for public research.

In the last decades, SNA has been used in various disciplines such as business [3], academics [4], politics [5], economics [6], law enforcement [7], sports [8], health care [9], and daily life activities that are highly related to real-world problems. SNA is most commonly applied to help to improve the effectiveness and efficiency of decision-making processes and deals with different issues. Few of them are very popular in SNA research: Team formation, Link prediction, Leadership detection, Community detection, Migration Between Communities, Sentimental Analysis, Collaborative Recommendation, Influence Analysis, and Fraud Detection.

This dissertation focuses on the formation of teams and the prediction of future interactions in social networks.

1.1.1 Forming Successful Teams

Today, it is essential to have some collective thoughts and creative ideas for productive results in various fields such as academic collaborations, educational, healthcare, industrial organizations, human resource management settings. As specialization in every field increases, there is a need for an expert in specific skill. Therefore, bringing these experts together as a team for a collaborative working environment would be an excellent idea for the effective result. The combined expertise of a group can usually produce results considerably surpassing the sum of the individual capabilities. However, what guaranty do we have that the combination of these individuals who

[4]https://www.linkedin.com/

may or may not know each other would actually succeed? The links connecting these individuals are very critical. Are these connections between the individuals that we connected as a team with the right people? Will they make a huge difference in the outcomes of the result? Each individual has his or her demands and objectives that might conflict with team objectives. Creating and organizing such teams is highly challenging and not a straightforward task.

Team Formation Problem (TFP) has been tried to tackle by the researchers from various disciplines for a long time. Earlier, research from psychology and sociology tried to understand what control the individual and social behaviors of members of a team have. Then, because human natures are unpredictable, the mathematicians and statisticians tried to approach this problem in the dynamic environment. Zkarian *et al.* [10] designed a conceptual framework for the selection of multi-functional teams. Although the general idea of TFP is highly connected with social sciences, social scientists have not designed any effective model to solve TFP. Therefore, Lappas *et al.* [11] incorporated social networks with TFP to discover the team of experts for the first time in SNA. Inspired by his work, TFP has taken a great deal of attention from various researchers [12, 13, 14, 15].

1.1.2 Predicting Future Collaboration

The individual of a social network often has insufficient information about the existing other individuals with which future interaction might prove fruitful. Moreover, even with the presence of such information, predicting in advance which potential collaboration should pursue is a highly challenging task [16]. The process of detecting suitable members for teams in a social network is typically a real-time application of link prediction.

Although the link prediction problem has been discussed over the last two decades, the work of Jon Kleinberg and David Liben-Nowell [17] has drawn a great deal of at-

tention in recent years. Later, numerous methods have been proposed to solve the link prediction problem because it has several applications including friend recommendation [18], classify the behavior and motion of people [19], and disease gene prediction [20]. It has been solved in two different settings: static and dynamic network settings. The link prediction in static network considers a single snapshot, G_t of a network at time t and is used to determine new links in time $t'(> t)$, while the dynamic networks considers sequence of snapshots, $\{G_1, G_2, \ldots, G_{t-1}\}$ of an evolving network in $\{1, 2, \ldots, t-1\}$ and are used to determine new links in time t. The traditional approaches consider the topological structures of the network and measure the similarity metrics using various methods including Common neighbors (CN) [21], Adamic Adar (AA) [18], and Resource Allocation (RA) [22]. Later, many machine learning methods have been proposed to improve the accuracy and to handle the complexity such as DeepWalk [23], LINE [24] and Node2Vec [25] and deep learning techniques [26, 27, 28].

1.2 Motivations and Objectives

Team formation is not a straightforward process because each individual has his or her own demands and objectives that might conflict with team objectives. Defining TFP in different domains is challenging and not the same procedure. Another critical problem in TFP is formulating the cost functions to measure the performance of teams. The TFP has been proven to be an NP-Hard problem [11]. Many existing methods considered the entire social networks for the TFP to recommend effective teams of experts to complete given tasks under some constraints. They used several cost functions such as communication cost, expertise level, and workload level as a static score. On the other hand, considering the whole network together is not as effective as considering the evolving nature of networks. In real world, social networks

are changing over time; people join the networks and interact with existing other people to build strong relationships. Many existing models failed to bring dynamic nature with cost functions.

Problem 1.1. ***(Team Discovery)*** *Given a set of projects $\mathcal{P}$, a set of experts $\mathcal{V}$, and a social network that is modeled as graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, the problem of team discovery in social networks is to find a teams of experts $\mathcal{T}$ for $\mathcal{P}$ from $\mathcal{G}$ so that the evaluation costs of T is minimized or maximized.*

Predicting future interaction is another significant problem in TFP. Existing teams can collaborate or interact with new individuals for various purposes, such as to fulfill the required skill for the given tasks and new faculty contributing his skills with existing academic teams. It is an application of link prediction problem in social networks. The existing methods suffer from the accuracy of various network types and struggle with the size of the networks.

Problem 1.2. ***(Predicting link for Future Collaboration)*** *Given a sequence of snapshots $\{\mathcal{G}_{t-k}, \mathcal{G}_{t-k+1}, \ldots, \mathcal{G}_t\}$ of an evolving graph $\mathcal{G}$, with length k have the corresponding adjacency matrices $\{A_{t-k}, A_{t-k+1}, \ldots, A_t\}$, the primary objective of link prediction in dynamic networks is to model a framework to learn the following function to predict the topological changes, mainly in links at time $t+1$.*

$$A_{(t+1)} = f(A_{t-k}, A_{t-k+1}, , \ldots, A_t) \tag{1.1}$$

where $f(A_{t-k}, A_{t-k+1}, , \ldots, A_t)$ represents the model required to predict the adjacency matrix $A_{(t+1)}$ at time $t+1$.

Motivated by the observations that sections 1.1.1 and 1.1.2 elaborate, this dissertation addresses the above research problems in social networks. The primary objective of this dissertation is to model a computational framework to discover teams of experts in various applications and predict the potential for collaboration in the future from a given dynamic social network.

1.3 Research Contributions

This research examines the team formation problem in three different applications: academic collaborations, health care settings, and industrial-organizational settings, and proposes new computational frameworks by using a knowledge-based cultural algorithm, which was introduced by Reynolds [29]. TFP can be formulated as an optimization problem where our goal is to maximize or minimize the cost functions in order to discover the teams of experts to complete given tasks under certain constraints. The significant difference that this research work has from other existing works is utilizing different sources of knowledge from the network and handling the searching process smoothly. It helps to address the computational challenges in TFP. Another major contribution for the first time in healthcare is to model a framework for palliative care multidisciplinary networks to identify and measure the performance of the optimal team of care providers in highly dynamic and unbalanced social networks of volunteer, community, and professional caregivers. This study also designs a visualization framework to explore and monitor the evolution in the structure of the care networks. It helps to identify the isolated patients, imbalanced resource allocation, and uneven service distribution in the network. This contribution is acknowledged by Hospice and the Windsor Essex Compassion Care Community in partnership with the Faculty of Nursing.

Another contribution of this research is to model a unified framework for TFP by involving essential cost functions. For that purpose, the temporal nature of both communication cost and expertise level is examined to introduce new score functions. To optimize the objective functions, the model uses the multi-objective cultural algorithms, which extract various sources of knowledge and update each generation. In each case of TFP, the experimental results reach closest to the exact algorithm and outperform well-known searching methods.

In order to predict future collaboration, this research focuses on the evolving pattern of vertices of a given network $\mathcal{G}$ over time. As another contribution, a time-varying score function is introduced to evaluate the activeness of nodes that uses the number of new interactions and the number of frequent interactions with existing connections. To consider the impact of timestamps of the interactions, the score function engages a time difference of the current time and the time of the interaction occurred. Thereafter, a probability function is introduced based on the activeness of the individual. This study includes two additional objective functions in our model: a weighted shortest distance between any two nodes and a weighted common neighbor index. The Multi-Layer Perceptron (MLP) is a deep learning architecture, used as a classifier to predict the link formation in the future and define our model as a binary classification problem. Experimental evaluation against network embedding and basic heuristic methods shows significant improvement and reaches up to 93%.

This dissertation attempts another contribution to predicting the links for future collaborations. The proposed model aims to improve the accuracy of the prediction as well as reduce the complexity. These are two major issues in the link prediction problem. This research uses common neighbors based subgraph of a target link and learns the transitional pattern of it for a given dynamic network. A set of heuristic features of the evolving subgraph is extracted to gather additional information about the target link. In this way, this research avoids examining the entire network. Additionally, this model uses some new mechanisms to reduce computational costs. It generates a lookup table to keep the required information of links of the network and uses a hashing method to store and fetch link information. An algorithm is introduced to construct feature matrices of the evolving subgraph to learn transitional link patterns. This model transforms the dynamic link prediction to a video classification problem and uses Convolutional Neural Networks with Long Short-Term Memory neural networks. To verify the effectiveness of our studies, extensive experiments are

carried out on five real-world dynamic networks. Those results are compared against four network embedding methods and basic heuristic methods and show significant improvement than other approaches.

1.4 The structure of the dissertation

The rest of this dissertation is organized as follows.

Chapter 2 elaborates on how to tackle the problem of finding a team that covers a set of required skills in academic collaborations networks. It uses two well-known functions to evaluate the communication cost of a team: the diameter and the sum of distances. It explains the benefits of utilizing a knowledge-based evolutionary optimization algorithm to solve this problem. This research creates a key path for the rest of the works in this dissertation.

Chapter 3 examines the high-level benefits of social network analysis in healthcare settings. It describes the palliative care multidisciplinary networks and how to solve the problem from the unbalanced social networks of volunteer, community, and professional caregivers by assigning optimal teams to serve patients. Chapter 4 illustrates the necessity of visualizing healthcare networks and their advantages.

Chapter 5 presents the framework for industrial organization settings. It explains how this model varies from other cases in the previous chapters. Industries focus on profits from a set of projects. This chapter elaborates on how to maximize profits by hiring efficient teams of experts under a given budget.

Chapter 6 provides complete descriptions of a unified framework for the team formation problem. It summarizes the existing team formation approaches, and investigates various cost functions to handle the dynamic social networks.

In chapters 7 and 8, the problem of predicting links for future collaboration in social networks is reviewed. Chapter 7 describes the impact of analyzing the active-

ness of vertices for link prediction in dynamic networks, while chapter 8 examines the method to handle the complexity of dynamic networks by using efficient subgraph extraction.

Finally, the conclusions of the dissertation along with our future directions is presented in Chapter 9.

Bibliography

[1] J. Scott and P. J. Carrington, *The SAGE handbook of social network analysis*. SAGE publications, 2011.

[2] A.-L. Barabâsi, H. Jeong, Z. Néda, E. Ravasz, A. Schubert, and T. Vicsek, "Evolution of the social network of scientific collaborations," *Physica A: Statistical mechanics and its applications*, vol. 311, no. 3-4, pp. 590–614, 2002.

[3] C. Dong and H. Rim, "Exploring nonprofit-business partnerships on twitter from a network perspective," *Public Relations Review*, vol. 45, no. 1, pp. 104–118, 2019.

[4] X. Kong, Y. Shi, S. Yu, J. Liu, and F. Xia, "Academic social networks: Modeling, analysis, mining and applications," *Journal of Network and Computer Applications*, 2019.

[5] L. A. Pal and J. Spence, "Event-focused network analysis: a case study of anti-corruption networks," *Policy and Society*, pp. 1–22, 2020.

[6] A. Polyakova, M. Loginov, A. Serebrennikova, and E. Thalassinos, "Design of a socio-economic processes monitoring system based on network analysis and big data," 2019.

[7] J. S. Hollywood, M. Vermeer, D. Woods, S. E. Goodison, B. A. Jackson, RAND,

and U. S. of America, "Using social media and social network analysis in law enforcement," 2018.

[8] F. Korte, "Network analysis in team sports," Ph.D. dissertation, Technische Universität München, 2019.

[9] N. S. Jessani, C. Babcock, S. Siddiqi, M. Davey-Rothwell, S. Ho, and D. R. Holtgrave, "Relationships between public health faculty and decision makers at four governmental levels: a social network analysis," *Evidence & Policy: A Journal of Research, Debate and Practice*, vol. 14, no. 3, pp. 499–522, 2018.

[10] A. Zzkarian and A. Kusiak, "Forming teams: an analytical approach," *IIE transactions*, vol. 31, no. 1, pp. 85–97, 1999.

[11] T. Lappas, K. Liu, and E. Terzi, "Finding a team of experts in social networks," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2009, pp. 467–476.

[12] C.-T. Li and M.-K. Shan, "Team formation for generalized tasks in expertise social networks," in *Social Computing (SocialCom), 2010 IEEE Second International Conference on*. IEEE, 2010, pp. 9–16.

[13] C. Dorn and S. Dustdar, "Composing near-optimal expert teams: a trade-off between skills and connectivity," in *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*. Springer, 2010, pp. 472–489.

[14] M. Kargar and A. An, "Discovering top-k teams of experts with/without a leader in social networks," in *Proceedings of the 20th ACM international conference on Information and knowledge management*. ACM, 2011, pp. 985–994.

[15] A. Gajewar and A. Das Sarma, "Multi-skill collaborative teams based on densest

subgraphs," in *Proceedings of the 2012 SIAM International Conference on Data Mining*. SIAM, 2012, pp. 165–176.

[16] M. Pavlov and R. Ichise, "Finding experts by link prediction in co-authorship networks." *FEWS*, vol. 290, pp. 42–55, 2007.

[17] D. Liben-Nowell and J. Kleinberg, "The link-prediction problem for social networks," *Journal of the American society for information science and technology*, vol. 58, no. 7, pp. 1019–1031, 2007.

[18] L. A. Adamic and E. Adar, "Friends and neighbors on the web," *Social networks*, vol. 25, no. 3, pp. 211–230, 2003.

[19] H. Sid Ahmed, B. Mohamed Faouzi, and J. Caelen, "Detection and classification of the behavior of people in an intelligent building by camera." *International Journal on Smart Sensing & Intelligent Systems*, vol. 6, no. 4, 2013.

[20] X. Wang, N. Gulbahce, and H. Yu, "Network-based methods for human disease gene prediction," *Briefings in functional genomics*, vol. 10, no. 5, pp. 280–293, 2011.

[21] M. E. Newman, "Clustering and preferential attachment in growing networks," *Physical review E*, vol. 64, no. 2, p. 025102, 2001.

[22] T. Zhou, L. Lü, and Y.-C. Zhang, "Predicting missing links via local information," *The European Physical Journal B*, vol. 71, no. 4, pp. 623–630, 2009.

[23] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2014, pp. 701–710.

[24] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "Line: Large-scale information network embedding," in *Proceedings of the 24th international con-*

ference on world wide web. International World Wide Web Conferences Steering Committee, 2015, pp. 1067–1077.

[25] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining.* ACM, 2016, pp. 855–864.

[26] X. Li, N. Du, H. Li, K. Li, J. Gao, and A. Zhang, "A deep learning approach to link prediction in dynamic networks," in *Proceedings of the 2014 SIAM International Conference on Data Mining.* SIAM, 2014, pp. 289–297.

[27] M. Rahman and M. Al Hasan, "Link prediction in dynamic networks using graphlet," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases.* Springer, 2016, pp. 394–409.

[28] J. Chen, J. Zhang, X. Xu, C. Fu, D. Zhang, Q. Zhang, and Q. Xuan, "E-lstm-d: A deep learning framework for dynamic network link prediction," *arXiv preprint arXiv:1902.08329*, 2019.

[29] R. G. Reynolds, "An introduction to cultural algorithms," in *Proceedings of the third annual conference on evolutionary programming.* World Scientific, 1994, pp. 131–139.

Chapter 2

A Knowledge-based Computational Algorithm for Discovering a Team of Experts in Social Networks

In this study, we examine the problem of finding a team that covers a set of required skills of a task in a given social network. The proposed model is suitable for solving team formation problems in various types of applications, including academic collaborations and educational settings. Because this problem has been proven to be an NP-hard problem [1], we attempt it using the knowledge-based cultural algorithms as a significant contribution. This research opens a path for the rest of the works in this dissertation.

2.1 Introduction

An expert network is a social network that contains professionals who have skills and expertise in particular areas. With the growth of the Internet, online social networks of experts have become popular and more enterprises seek to find talent and expertise from such networks to complete a project or task. Some examples of these expert

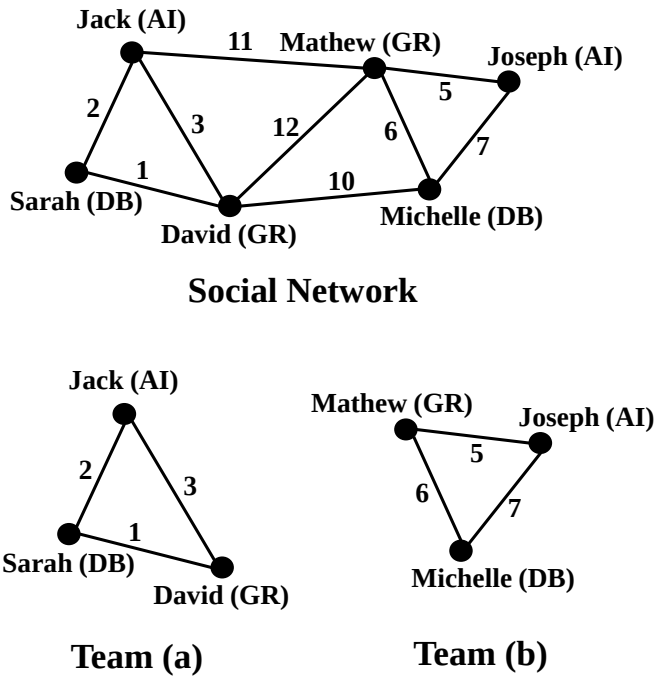


Figure 2.1: A social network and two teams to perform a project that requires artificial intelligence (AI), databases (DB), and graphics (GR). Edge weights show the communication cost between the experts. Smaller values indicate two experts communicate more efficiently and have more past experience.

networks are the employment-based service LinkedIn, the code repository hosting service GitHub, and the research-based author websites such as Google Scholar and DBLP. In such networks, a node is an expert that is associated with a person. Each expert possesses a set of skills that determine his/her expertise based on his/her education, past experience, or training. If two experts have past collaboration, they will be connected to each other in the social network. This includes writing a paper together, being a member of the same GitHub repository, or being a member of the same project in the same company. In some situations, there might be a different degree of collaboration among the experts. For example, two developers might participate in more than one GitHub repository. In this case, the weight of the edges determine the strength of the relationship between the two experts [2, 3, 4].

We focus our study on the problem of forming a team of experts from a given social network to complete a certain project. In this context, a project is composed

of a set of skills, where each skill should be covered by one expert in a team in order to complete the project. Traditionally, team formation has been studied in operation research and most of the classic approaches do not evaluate the underlying social network of experts. However, this is an important aspect of a successful team. The success of a project depends on how well the experts on the team communicate and collaborate with each other. If they have past experience, it is more likely that they finish the project in a timely manner. Thus, in addition to finding a set of experts that posses the required skills, we are interested in minimizing the communication cost between them as well.

To motivate the problem and illustrate the benefit of evaluating the communication cost, consider the social network presented in Figure 2.1. We have six experts in total. The expertise of each expert is also shown in the graph. The edge weight shows the communication cost between experts. Smaller values determine the experts have more past experience and therefore, will form a more collaborative team in the future. For example, the communication cost between Sarah and David is the smallest (i.e., it is one). Thus, we can conclude that Sarah and David will collaborate efficiently in the future. On the other hand, the communication cost between Mathew and David is 12. Hence, it is unlikely that they will form a strong team together.

Assume we need to perform a project that needs expertise in three areas: artificial intelligence (AI), databases (DB), and graphics (GR). Two teams to perform this project are shown in Figure 2.1. Each of these teams has three experts who together cover the three required skills. However, the overall communication cost among the experts in the team (a) is lower than the one in team (b). The edge weights in the team (a) are 1, 2, and 3; while they are 5, 6, and 7 in team (b). Therefore, team (a) would form a more successful team in terms of the communication cost. In the next section, we formally define two functions to calculate the communication cost among the experts for any given team.

This problem has been introduced for the first time by Lappas *et al.*, [1]. They propose two functions for estimating the communication cost of a team and prove that minimizing each function is an NP-hard problem. Therefore, they propose greedy algorithms to find the best approximate team. Kargar and An propose a new function (i.e., Sum of distances between each pair of experts), to measure the communication cost between experts of a team [5]. They suggest that the sum of distances is not only biased towards some of the experts but guarantees that all of the experts are equally involved in the calculation of the communication cost. They prove that minimizing this distance function is also NP-hard and they propose greedy algorithms to find approximate answers.

One limitation of these greedy algorithms is that they may not scale well when the size of the network expands. It makes selections based on what looks best at the moment and does not refine the solutions based on new information. In other terms, choices are locally optimum but not necessarily globally optimum. For finding the best team, they build a team around each skill holder. By increasing the number of skill holders, for each required skill, their run time increases. Another limitation is that they may not optimize the communication cost properly. The communication cost of the team returned by the greedy algorithms might be far from the best team obtained by the exhaustive search. To address these issues we propose the use of knowledge-driven population based evolutionary algorithms that have been proven to successfully solve complex optimization problems. We propose the use of a Cultural Algorithm (CA) as the evolutionary framework to search for the optimum team of experts in a social network.

We make the following contributions in this paper:

1. We propose a knowledge-based evolutionary optimization model to find a team of experts from a social network while minimizing the communication cost. To the best of our knowledge, there is no prior work to propose such knowledge-

based algorithm to solve this problem.

2. We compare our proposed algorithm with greedy, genetic, random, and exact algorithms. The random algorithm is designed to be used as the baseline. As the gold standard, and since the problem is NP-hard, we develop an exhaustive search algorithm to find the exact answers.

3. We analyze and compare different algorithms in terms of the communication cost, the run time, the number of iterations and the size of the population.

4. In our experiments, we use real world networks with 50K, 100K and 200K nodes derived from the DBLP dataset. DBLP is a network among authors of scientific papers in computer science.

The rest of the paper is organized as follows. In the next section, related work is discussed. We present problem statement in Section 2.3. In Section 3.3.1, we propose our knowledge-based framework to find the best team of experts. In Section 2.5, a comprehensive set of experiments over a real dataset is presented. Finally, we conclude in Section 2.6.

2.2 Related Work

There are two main related approaches to deal with the problem of finding teams of experts: Greedy algorithms and evolutionary-based methods. In the following paragraphs, we briefly review them.

Discovering a team of experts from a social network was first introduced by Lappas *et al.*, [1]. The authors proposed two communication cost functions. Li and Shan generalized this problem by associating each required skill with a specific number of experts [6]. Kargar and An proposed the sum of distance function to find the best team and argued that the new function is fairer to all team members for calculating the

communication cost [5]. Another communication cost function based on the density of the induced team subgraph was proposed in [7]. When dealing with multiple projects, authors of [8] minimized the maximum load of the experts. However, they ignored the communication cost among experts. Minimizing both communication cost and load balance was studied in [9] and [10].

Kargar *et al.*, proposed to find a team of experts to minimize the communication cost and the personnel cost of a team [11]. They assumed every expert is associated with a cost in order to perform a task in a given project. They merged the two objective functions into one using a tradeoff parameter. Authors of [12] solved this problem by finding the set of Pareto teams. They also found the best team minimizing the communication cost under the given personnel cost budget. Recently, Li *et al.*, proposed to find a replacement when a team member becomes unavailable [2].

The team formation problem is well explored by the operation research community. Different works use the genetic algorithms, simulated annealing and branch-bound, with the goal to find a team for performing the given task [13, 14, 15, 16]. All of these works ignore the social network and graph structure behind the experts and do not minimize the communication cost among team members.

Recently, Awal *et al.*, proposed to find a team of experts in social networks using a collective intelligence index [17]. They use the summation of the trust score and the expertise score of experts as the fitness function. To optimize it, they apply general genetic algorithm (GA), and use two random points for the crossover operation and mutate a random expert. The experiment of this work is however limited since it performs with synthetic dataset composed of only 30 experts. Furthermore, the results are not compared to any other non-evolutionary-based algorithms.

Wi *et al.*, studied the team formation problem in a research organization [18]. In their work, they evaluate two different selection methods to choose team members and project managers. The team members are selected using GA and the knowledge

competence score of the candidates for a certain project. Then, among selected team members, the project manager is chosen based on the knowledge competence score and social network measures. A fuzzy inference system is used to calculate knowledge competence. The proposed algorithm is evaluated in a research organization with only 45 researchers.

Blas *et al.*, studied team formation based on group technology and propose a hybrid grouping genetic algorithm to find the best team [19]. The number of groups vary from one to several groups. The population is selected by rank-based wheel-selection mechanism. Furthermore, the fitness function is defined based on the position of the individual in the ranking mechanism. In this algorithm, two-crossover points are used to produce a new offspring. The mutation is designed by reordering groups with random selection. The algorithm is evaluated with 61 lecturers and 65 courses at a Spanish university.

Ani *et al.*, proposed a method for group formation using a genetic algorithm [20]. Their proposed fitness function estimates the minimum number of good, moderate, and poor students in a group. One-point crossover is used to implement a new offspring. Mutation is done by exchanging an arbitrary bit in the genetic sequence with its original state by generating a random variable for each bit. The authors use 35 students in their experiments.

Authors of [21] presented a social matching model to bring the skill holders together in which the skill holders support the composition of the teamwork. The proposed fitness function calculates an individual's aptitude and profiles, and sociometric team cohesion. Both roulette-wheel selection and elitism methods are applied to select the best individuals. Their approach use one cut-off point to generate children using crossover operator of 60 to 65 percent of the population. For the mutation, a rate of 0.5% to 5.0% is used where two genes are randomly selected. Then, all other genes are shifted from the first location to the second location and then the second

location is filled by the first gene.

In this paper, we propose a custom CA which is a knowledge-driven population-based evolutionary algorithm to minimize the communication cost. We compare the results to a state of the art greedy algorithm and a genetic algorithm over a real world DBLP dataset with 200K experts. We show the advantage of using CA to optimize the fitness function and scalability of the framework. The main difference between our approach and the reviewed approaches is using the extracted knowledge from the social network in the optimization process. We also use two well-known fitness functions to evaluate the communication cost of the teams.

2.3 Problem Statement

Let $A = \{a_1, a_2, \ldots, a_n\}$ be a set of n experts, and $K = \{k_1, k_2, \ldots, k_m\}$ be a set of m skills. Each expert a_i has a set of skills, specified as $S(a_i)$, and $S(a_i) \subseteq K$. If $k_j \in S(a_i)$, expert a_i posses skill k_j. A subset of experts $A' \subseteq A$ have skill k_j if at least one of them posses k_j. For each skill k_j, the set of all experts that posses skill k_j is specified as $A(k_j) = \{a_i | k_j \in S(a_i)\}$. A project $P = \{k_1, k_2, \ldots, k_s\}$ is composed of a set of s skills that are required to be completed by some experts. A subset of experts $A'' \subseteq A$ is able to *complete* a project P if $\forall k_j \in P \; \exists \; a_i \in A'', k_j \in S(a_i)$.

An underlying social network connects the experts in A. This social network is modeled as an undirected graph G. Each expert in A is represented as a node in graph G. Terms *node* and *expert* might be used interchangeably in this work. If the experts have past collaboration, their associated nodes in G are connected together by an edge. If different levels of past collaboration between two experts are taken into account, then the input graph G is weighted. In this case, the smaller the edge weight between two experts, the two experts had more collaboration in the past and will collaborate more efficiently in the future. The distance between two experts a_i

and a_j, specified as $dist(a_i, a_j)$, is equal to the sum of the weights on the shortest path between them in the input graph G. If a_i and a_j are not connected in graph G (i.e., there is no path between a_i and a_j in G), the distance between them is set to ∞. Next, we define a team of experts and the problem we tackle in this work.

Definition 2.1. ***(Team of Experts)*** *Given a set of experts A and a project P that needs a set of skills $\{k_1, k_2, \ldots, k_s\}$, a* team of experts *for P is a set of s skill-expert pairs:*

$\{\langle k_1, a_{k_1}\rangle, \langle k_2, a_{k_2}\rangle, \ldots, \langle k_s, a_{k_s}\rangle\}$,

where a_{k_j} is an expert that posses skill k_j for $j = 1, \ldots, s$. This means expert a_{k_j} is responsible for skill k_j.

Based on this definition, one expert is assigned to each skill k_j. In other words, each skill is covered by one expert, but one expert may be assigned to more than one skill. For each project P, there might be many teams that are able to complete the required skills. However, we are interested in teams that are able to communicate with each other effectively. Therefore, we use two functions to evaluate the communication cost among experts of a team.

Definition 2.2. ***(Diameter)*** *Given a graph G and a team of experts $\{\langle k_1, a_{k_1}\rangle, \langle k_2, a_{k_2}\rangle, \ldots, \langle k_s, a_{k_s}\rangle\}$, the* diameter *of this team is the largest shortest distance between any two experts a_{k_i} and a_{k_j} for $1 \leq i < j \leq s$.*

For example, the diameter of teams (a) and (b) in Figure 1 is 3 and 7 respectively. Therefore, team (a) has lower communication cost in terms of the diameter function. The diameter function is first proposed in [1].

Definition 2.3. ***(Sum of Distances)*** *Given a graph G and a team of experts $\{\langle k_1, a_{k_1}\rangle, \langle k_2, a_{k_2}\rangle, \ldots, \langle k_s, a_{k_s}\rangle\}$, the* sum of distances *of the team is defined as*

$$sumDistance = \sum_{i=1}^{s} \sum_{j=i+1}^{s} dist(a_{k_i}, a_{k_j})$$

where $dist(a_{k_i}, a_{k_j})$ is the distance between a_{k_i} and a_{k_j} in G (i.e., the sum of weights on the shortest path between a_{k_i} and a_{k_j}).

For example, the sum of distances of teams (a) and (b) in Figure 1 is 6 and 18 respectively. Clearly, team (a) is more collaborative than team (b) and its sum of distances is much smaller than the one for team (b). The sum of distances function is first proposed in [5]. Both of the diameter and the sum of distance functions measure the communication cost of the team.

Problem 2.1. ***(Team Discovery)*** *Given a project P, a set of experts A, and a social network that is modeled as graph G, the problem of team discovery in social networks is to find a team of experts T for P from G so that the communication cost of T, defined as the diameter of T or the sum of distances of T, is minimized.*

Finding teams of experts minimizing the diameter or the sum of distances of team T is proved to be NP-hard in [1] and [5] respectively. Therefore, authors of [1] and [5] propose greedy algorithms to find the best approximate teams minimizing these functions. The idea of these algorithms is as follows.

For finding the best approximate team, the algorithms build a tree around one of the skill holders. Each skill holder a_r (a_r is an expert that posses at least one of the required skills) in the graph is considered as a potential root for an answer tree. To build a tree around a_r, for each given skill k_i, the closest node a_{k_i} is assigned to a tree rooted at a_r. For finding the team with the smallest diameter, the tree in which its maximum edge has the lowest value among all other trees is chosen as the best approximate team. The tree with the lowest sum of the edge weights is the best approximate team regarding the sum of distances function. However, and as we show in the experiments, these algorithms do not scale well when the size of the graph or the frequency of skill holders increase. Also, our proposed algorithm outperforms both of these greedy algorithms (and also the genetic algorithm) in terms of the communication cost.

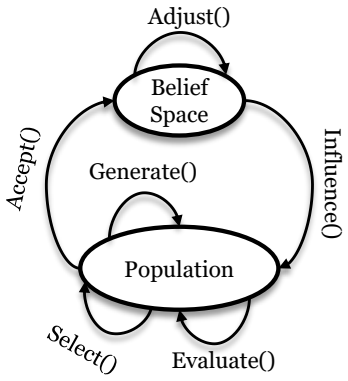


Figure 2.2: The knowledge-based framework for finding best answer

2.4 Algorithm

In this section, we propose a knowledge-based evolutionary method based on the cultural algorithm (CA) to find the best team of experts. As shown in Figure 2.2, our framework is a dual inheritance system consists of population and belief spaces which continually co-evolve each other during the optimization process [22, 23].

In this framework, the best of team of experts is found by executing a series of iterations. The population space is used to maintain the current list of generated teams in each iteration. We start by producing a predefined number of random teams to make the initial population (team representation is discussed in Section 2.4.1). The belief space (to be discussed in Section 2.4.2) is an extension unit which captures the extracted knowledge from the population. The quality of each team is evaluated by its communication cost (i.e., diameter or sum of distances). The new population is generated from a combination of top-teams in current iteration, teams that are produced from belief space, and teams that are generated from standard genetic algorithm operations (i.e., crossover and mutation). The main idea is that the obtained knowledge guides the search direction and evolves the teams faster than basic genetic evolution. Note that unlike genetic evolution, in our implementation, the duplicated individuals are removed from the search space. Although this strategy

increases the run time of the algorithm, it produces teams with smaller communication cost.

Algorithm 2 is our solution to Problem 2.1 to find the best team of experts using the above framework. In lines 1 to 4, initial parameters are set. In line 5, the initial population which is composed of a list of randomly generated sp teams are created (sp is the size of population in each iteration). Note that this list is duplication free. The for loop of line 6 is executed ni times (ni is number of iterations). The for loop of line 8 calculates the communication cost of each team. In line 9, the current population is sorted based on their communication cost. We initialize the new population in line 10. The first el teams of the current population is moved to the new population in line 11 (el is a parameter that determines how many elite teams should be moved to the next generation). In lines 12 and 13, we create the belief space (Section 2.4.2). The while loop of line 15 generates the remaining of teams (other than the one from elite group) for next generation. We start by generating teams from belief space with 80% probability. We tested other values but 80% produced the best result, hence we use this value. We also use standard genetic algorithm operations (crossover and mutation) to generate teams with 20% probability. Note that we do not add duplicate teams to the new population (line 24). The best team is the first team of the last generation and is returned in line 28. Next, we describe three main components of the algorithm: individual representation, the belief space structure, and calculating communication cost function.

2.4.1 Team Representation

A team (individual) in this work represents a candidate solution to the problem. It consists of an array structure with s cells, which is the number of the required skills to complete a project. For example, as shown in Figure 2.3, if a project needs three skills, the size of the array, s, is equal to three. The value of each cell is selected from

Algorithm 1 Knowledge-based Team Discovery Algorithm

Input: graph G; input project as list of s required skills, $\{k_1, k_2, \ldots, k_s\}$; set of skills of each expert a_i, $S(a_i)$
Output: best team

```
1: ni ← number of iterations
2: sp ← size of population (number of teams in each iteration)
3: kb ← number of teams to build knowledge-based belief space
4: el ← number of elite teams for next generation
5: T[1...sp] ← initialize and generate first duplication free population
6: for i ← 1 ni do
7:    for j ← 1 sp do
8:       calculate communication cost of T[j]
9:    end for
10:   sort T[1...sp] based on communication cost
11:   TN[1...sp] ← initialize new population
12:   TN[1...el] ← T[1...el]
13:   SP ← T[1...kb]
14:   BS ← SP^T
15:   size ← el + 1
16:   while size ≤ sp do
17:      team ← initialize new team
18:      if rand1() ≤ 80% then
19:         team ← generate new team from belief space BS
20:      else
21:         if rand2() ≤ 80% then
22:            team ← generate new team using crossover strategy
23:         else
24:            team ← generate new team using mutation strategy
25:         end if
26:      end if
27:      if team ∉ TN then
28:         TN[size] ← team
29:         size ← size + 1
30:      end if
31:   end while
32:   T ← TN
33: end for
34: return T[1]
```

the set of experts which possesses that skill. Therefore, for each cell representing a required skill k_j, a value is filled by an expert, a_i, where $k_j \in S(a_i)$.

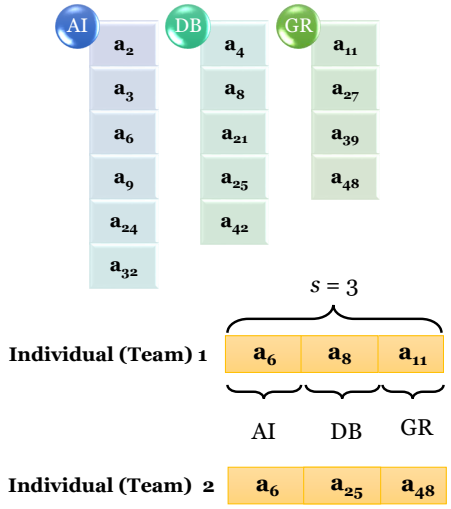


Figure 2.3: Individual representation

2.4.2 Belief Space

The belief space is defined as the transpose matrix of the selected population composed by the selected teams. Let a selected team (a team which is selected to change the belief space) be defined as $SI_j = [a^j_{k_1}, a^j_{k_2}, \ldots, a^j_{k_s}]$, where $k_i \in P$ and $a^j_{k_i} \in A(k_i)$, and $1 \leq i \leq s$. Now, let the selected population in each iteration with the size of t be defined as follows:

$$SP = \begin{bmatrix} SI_1 \\ SI_2 \\ \vdots \\ SI_t \end{bmatrix} = \begin{bmatrix} a^1_{k_1}, a^1_{k_2}, \ldots, a^1_{k_s} \\ a^2_{k_1}, a^2_{k_2}, \ldots, a^2_{k_s} \\ \vdots \\ a^t_{k_1}, a^t_{k_2}, \ldots, a^t_{k_s} \end{bmatrix}$$

Thus, the belief space is defined as $BS = SP^T$:

$$BS = \begin{bmatrix} a^1_{k_1}, a^2_{k_1}, \ldots, a^t_{k_1} \\ a^1_{k_2}, a^2_{k_2}, \ldots, a^t_{k_2} \\ \vdots \\ a^1_{k_s}, a^2_{k_s}, \ldots, a^t_{k_s} \end{bmatrix}$$

In other words, for each skill (a row of the BS matrix), the BS matrix shows the list of experts who have been appeared in the best teams. Assuming the optimal solution can be generated by combining the elements of the best teams, in the next iteration, the algorithm makes the new teams based on this belief space and not the actual set of experts for each skill, $S(k_j)$. This approach produces teams with lower communication costs and reduces the size of the search space.

2.4.3 Calculating Communication Cost

For calculating each of the communication cost functions (also called fitness functions), we frequently need to find the value of the shortest path between many pairs of nodes. However, computing the shortest path on-the-fly is very slow, while precomputing the shortest path values between all pairs of nodes takes too much space. Indeed, it takes $O(N^2)$ for a graph with N nodes. This quickly runs out of memory for big graphs. Therefore, we use an efficient indexing method called 2-hop cover [24, 25], which is a middle ground between the two extreme solutions that are discussed above. This efficient indexing technique returns the value of the shortest path between any pair of nodes in graphs with hundreds and thousands of nodes almost instantly (i.e., less than 10 μ seconds for the DBLP graph with 200K nodes).

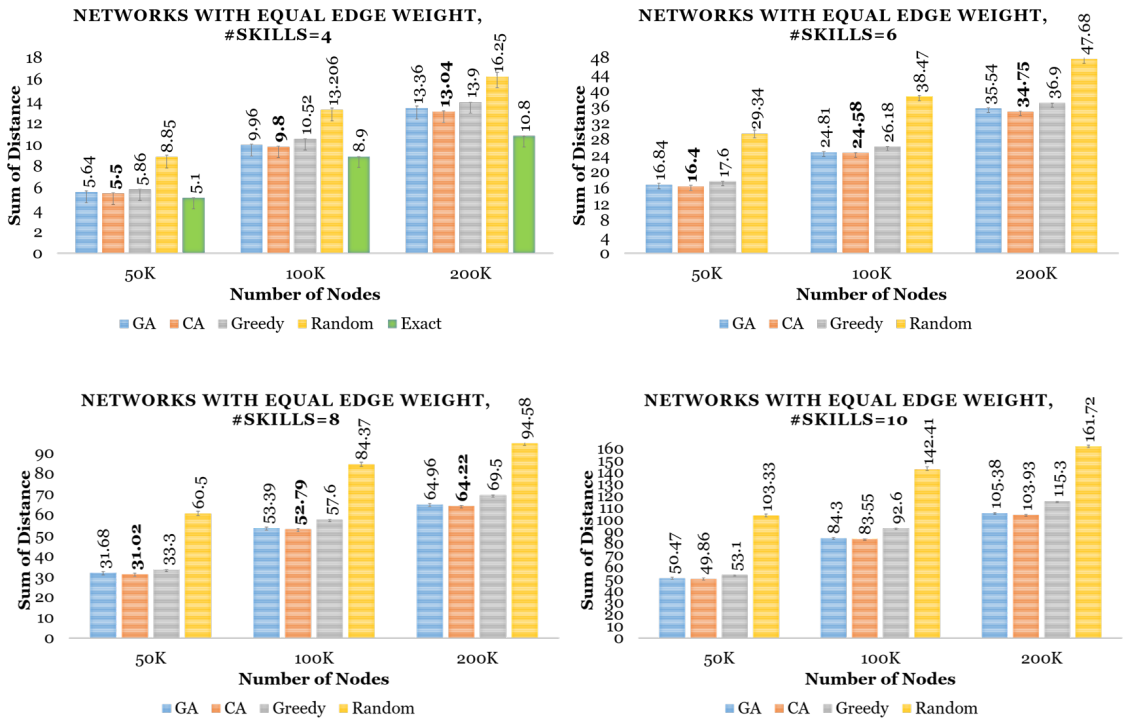


Figure 2.4: Comparison of the algorithms using the sum of distances for various networks and skills.

2.5 Experiments

In this section, we evaluate the performance of our proposed cultural algorithm (CA) over the various networks based on DBLP graph. For the fitness function, we use two communication cost functions: diameter and sum of distances. Since the sum of distances function is fairer towards all of the skill holders than the diameter function [5], and due to the space limit, most of our experiments are performed over the sum of distances. In addition, we compare our algorithm with the greedy algorithm proposed in [1] and [5]. We also develop a genetic algorithm (GA) and compare the results of our CA with it. The genetic algorithm is the modified version of the algorithm presented in [17]. In fact, the authors of [17] used a different fitness function to find the best team. Moreover, as a baseline for the comparisons, we compare the results with the random method, which simply select the team with the lowest communication cost

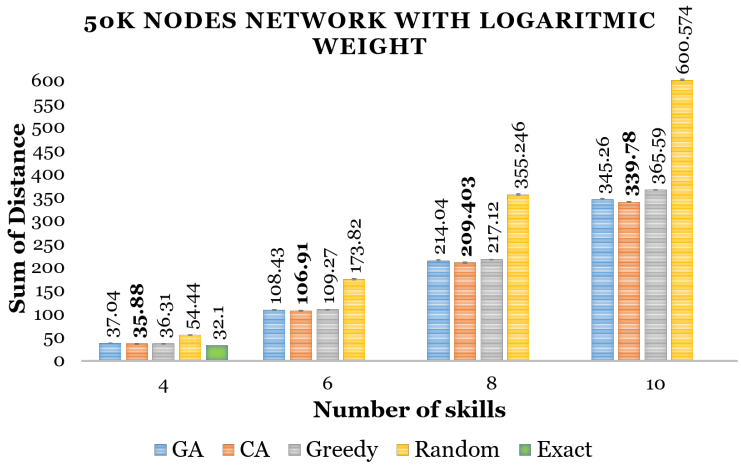


Figure 2.5: Comparison of the algorithms on a 50K nodes network with logarithmic edge weight.

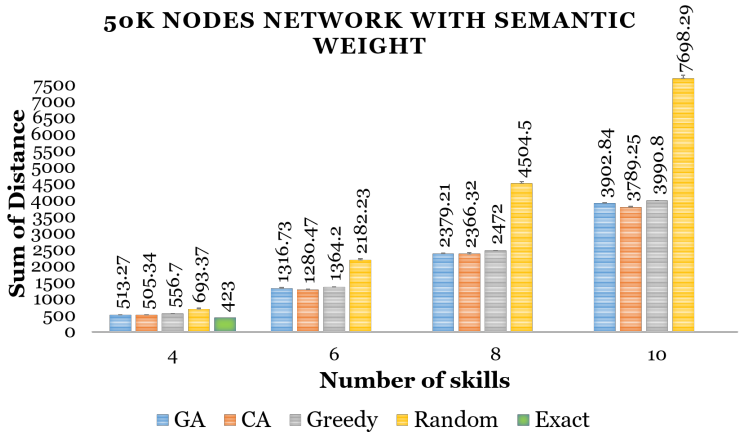


Figure 2.6: Comparison of the algorithms on a 50K nodes network with semantic edge weight.

among 10,000 random teams. We also compare the proposed algorithm with the exact algorithm in which the results are obtained using exhaustive search.

We create the input graph from the DBLP[1] dataset in the same way as [1] and [11]. The graph contains 200K nodes and 1.16M edges. In this case, the experts are authors of the papers. If two authors publish papers together, there will be an edge between them in the graph. The expertise of an expert (i.e., author), is extracted from the titles of his/her papers. For this study, we also use two sub-graphs of the

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

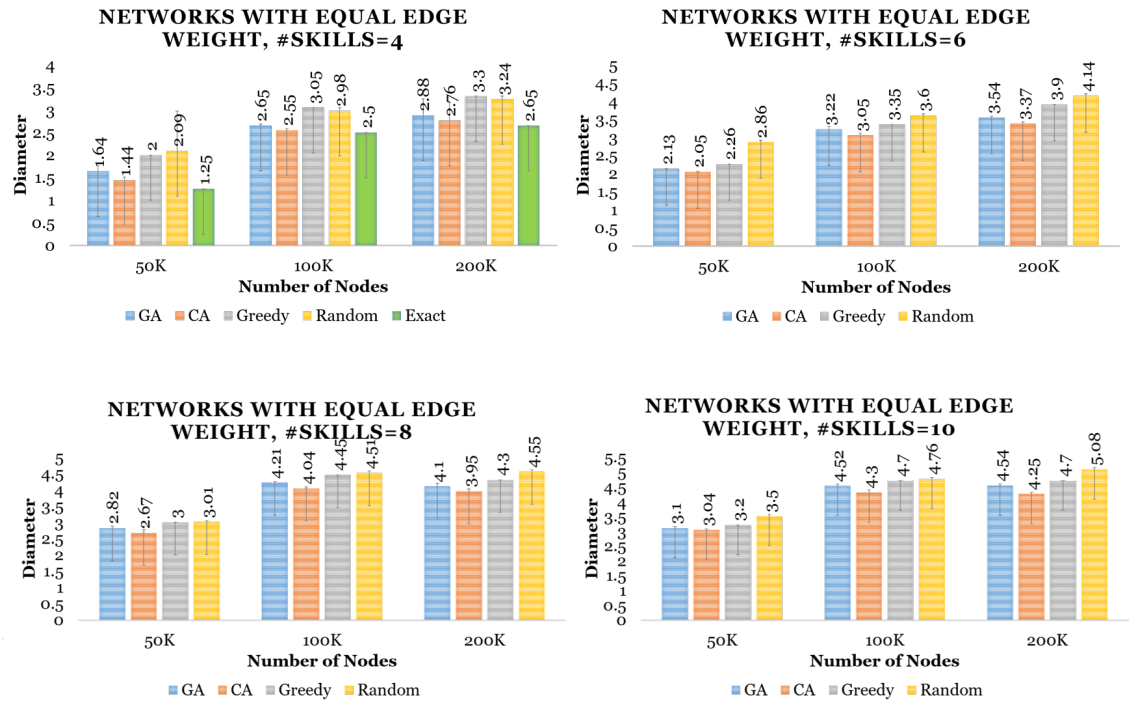


Figure 2.7: Comparison of the algorithms using the diameter for various networks and skills.

main graph with 100K and 50K nodes.

By changing the edge weighting mechanism as follows, we generate 3 various networks with 50K nodes.

- **Equal** weights, in which all edges have the weight of 1.0.
- **Logarithmic** weights, in which the weight of an edge between two experts a_i and a_j is $(\log_2(1 + deg(a_i) + \log_2(1 + deg(a_j))/2$, where $deg(a_i)$ is the degree of node a_i.
- **Semantic** weights, in which the weight of an edge is defined based on the number of co-authored publications.

Since we are interested in minimizing the communication cost, the semantic weights are reverted. In other words, the more publications two authors have with each other, the smaller the semantic edge weight between them.

As mentioned in the previous sections, each project is composed of a set of skills that need to be covered by experts. In our experiments, these skills are varying from 4 to 10 and for each one, 100 projects are randomly generated. Hence, a team with the lowest communication cost in each project is selected by the algorithms. The average value of the cost of the selected teams in these projects then is considered as the average fitness value of the algorithms. Therefore, the algorithm with the lowest average fitness value is considered as the best algorithm among the others. Note that exhaustive search takes very long time as the problem is NP-hard and the search space is exponential. Indeed, we could only run the exhaustive search for 4 required skills due to time and memory constraints.

Moreover, the skill frequency is the number of experts who posses that skill. For example, if the required skill is "databases" and its frequency is 150, it means there exist 150 experts in the network with expertise in "databases". Of course different skills have different frequencies. We study the effect of different skill frequency on the performance of algorithms. In this paper, when not stated, the skill frequency varies from 25 to 200.

For the CA and GA, the number of iterations and the size of the population is set to 100 and 300 respectively. All of the algorithms are implemented in Java and executed on an Intel Core i7 3.4 GHz computer with 16 GB of RAM.

2.5.1 Evaluation of Communication Cost Function

The experiments start by comparing the values of the communication cost functions (i.e. fitness function) obtained by the algorithms. Figure. 2.4, shows the values when the fitness function is the sum of distances for 50K, 100K and 200K equal weighted graphs based on the number of required skills. Note that the results of the exact algorithm are only available when the number of the required skills is 4 as we run out of memory for more than 4 number of required skills. The obtained results from the

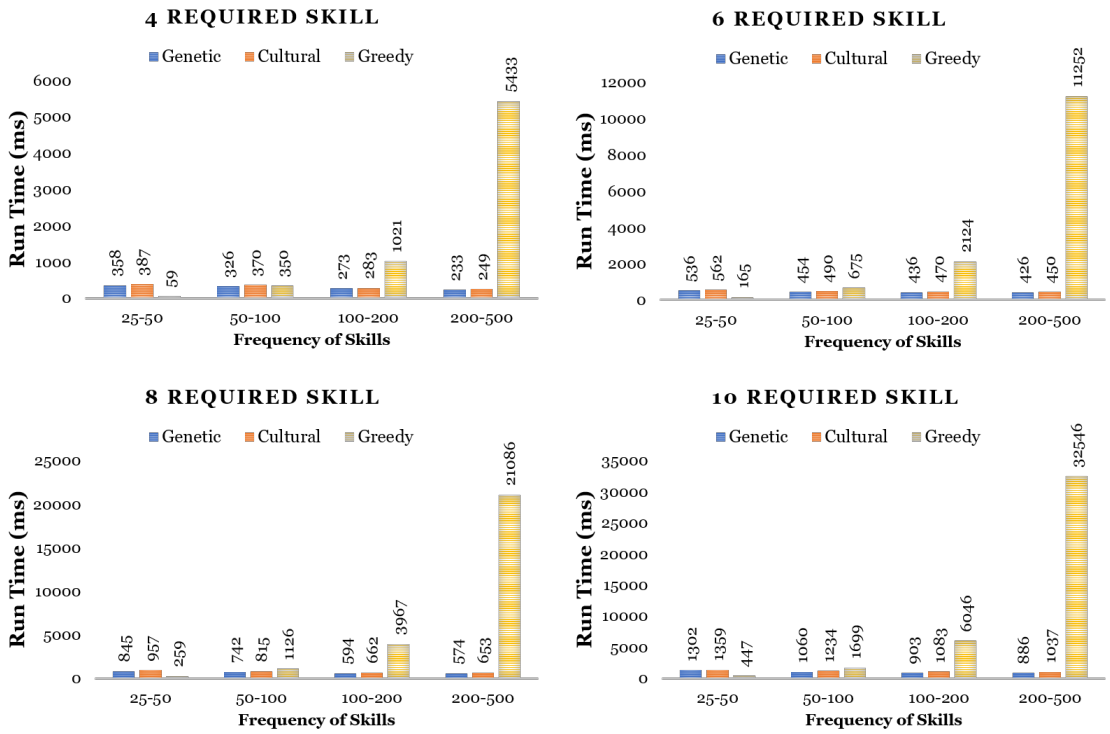


Figure 2.8: Run times of the algorithms with different numbers of required skills when the frequency of skills varies. The input graph contains 200K nodes.

same fitness function on 50K graphs with logarithmic and semantic weights also are shown in Figure. 2.5 and Figure. 2.6 respectively.

The results suggest that in almost all the cases, our proposed CA outperforms the greedy, genetic and random algorithms. When the number of required skills is 4, the results of the CA are the closest to the exact algorithm. Figure. 2.7 shows the same trend using the diameter as the fitness function. Due to the space limit, we only demonstrate the results on the 50K, 100K and 200K equal weight graphs. Similar to the sum of distances, our proposed CA outperforms greedy, genetic, and random algorithms. Meanwhile, the obtained results by CA is relatively very close to the available exact values which shows the high accuracy of our proposed algorithm.

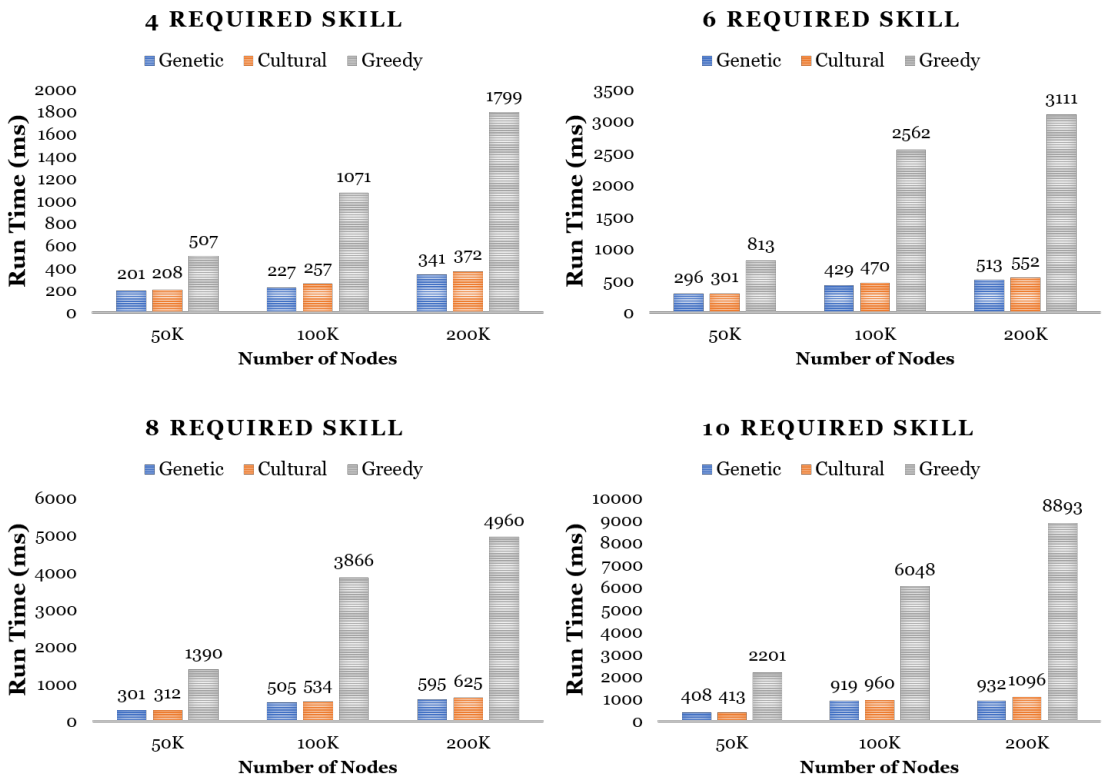


Figure 2.9: Run times of the algorithms with different numbers of required skills for the various size of the graphs. The frequency of skills is between 100 to 200.

2.5.2 Run Time

In this section, the run time of the algorithms is evaluated based on multiple parameters. Due to the space limit, only the obtained results for equal edge weights graphs are presented here. However, our experiments show that, the trend is the same for the graphs with semantic and logarithmic weights.

Figure. 2.8 shows the run time when the frequency of skills change. As mentioned before, the frequency of skills is the number of experts that hold a particular skill. For each range of required skills (e.g., 50-100), we create a bucket. This bucket contains all the skills in which each of them is possessed by the number of experts in the given range (e.g., 50 to 100 experts). Then, from this bucket, we choose the required number of skills (e.g., 6 skills), randomly.

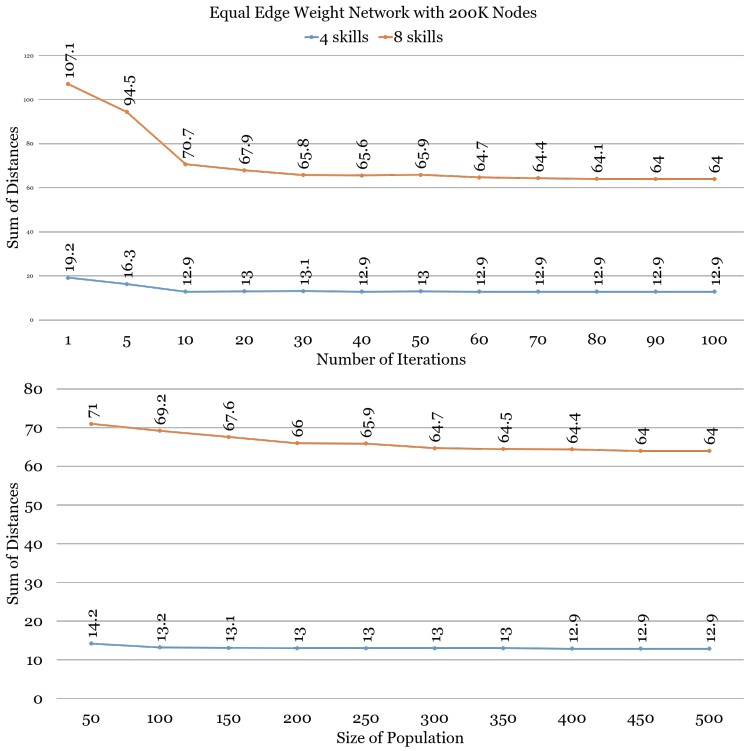


Figure 2.10: Obtained sum of distances (communication costs) with different numbers of iterations and population sizes on the 200K nodes network.

According to the results, the run times of GA and CA do not increase significantly when the frequency of required skills increases which is in contrast to the greedy algorithm. By increasing the frequency of skills, the run time of the greedy algorithm increases dramatically. This is expected because for each required skill, the algorithm checks all skill holders to find the one that is the closest to the current root. The GA runs a bit faster than the CA due to the process of removing duplicate genes in each iteration. However, the overall time is almost the same for the GA and CA.

Figure. 2.9 shows the obtained run times when the size of the graph changes. Similar to the previous experiments, no significant change is observed in the CA and GA, by increasing the size of the input graph. On the other hand, this is not the case for the greedy algorithm. Overall, the results suggest that both the CA and GA, scale well with different frequency of skills, different graph sizes and the different number of required skills.

2.5.3 Tuning the Evolutionary Optimization Algorithms

In this section, we examine the impact of the size of population and the number of iterations on the convergence of the CA. Figure. 2.10 illustrates the obtained fitness values by our proposed algorithm for 4 and 8 required skills when the iteration number varies from 0 to 100 and the fitness function is the sum of distances. According to the result, the fitness value does not significantly change after the 30th iteration. In addition, as shown in the figure the population of 300 produces near optimal results. However, increasing the size of the population might lead to slightly better results with the cost of increasing the run time.

2.6 Conclusion

In this paper we examined the problem of finding a team of experts in a social network that covers the set of required skills with the minimum communication cost among team members. To estimate the cost, two well-known functions have been employed based on the diameter and the distance between the edges. Since the twofold minimization problem is proven to be NP-hard, we proposed a knowledge based Cultural Algorithm to find the best collaborative team with least communication cost. We compared the results of our algorithm with state of the art greedy and genetic algorithms. The results suggest that the proposed CA can identify teams with smaller communication cost than greedy and the GA.

Furthermore, the communication cost of teams that are identified by our algorithm is the closest to the exact results obtained by the exhaustive search. Our results also suggest that in contrast to the greedy algorithm, CA and GA scale well with respect to the different frequency of skills. In the future, we plan to explore how to incorporate more constraints, such as the personnel cost and expertise level of experts, in our knowledge-based team formation computational model.

Bibliography

[1] T. Lappas, K. Liu, and E. Terzi, "Finding a team of experts in social networks," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining.* ACM, 2009, pp. 467–476.

[2] L. Li, H. Tong, N. Cao, K. Ehrlich, Y. Lin, and N. Buchler, "Replacing the Irreplaceable: Fast Algorithms for Team Member Recommendation," in *WWW*, 2015, pp. 636–646.

[3] S. B. Roy, L. Lakshmanan, and R. Liu, "From Group Recommendations to Group Formation," in *SIGMOD*, 2015, pp. 1603–1616.

[4] S. S. Rangapuram, T. Bühler, and M. Hein, "Towards realistic team formation in social networks based on densest subgraphs," in *Proceedings of the 22nd international conference on World Wide Web.* ACM, 2013, pp. 1077–1088.

[5] M. Kargar and A. An, "Discovering top-k teams of experts with/without a leader in social networks," in *Proceedings of the 20th ACM international conference on Information and knowledge management.* ACM, 2011, pp. 985–994.

[6] C.-T. Li and M.-K. Shan, "Team formation for generalized tasks in expertise social networks," in *Social Computing (SocialCom), 2010 IEEE Second International Conference on.* IEEE, 2010, pp. 9–16.

[7] A. Gajewar and A. D. Sarma, "Multi skill collaborative teams based on densest subgraphs," in *Proc. of the SDM'12*, 2012.

[8] A. Anagnostopoulos, L. Becchetti, C. Castillo, A. Gionis, and S. Leonardi., "Power in unity: Forming teams in large-scale community systems," in *Proc. of CIKM'10*, 2010.

[9] A. Anagnostopoulos, L. Becchetti, C. Castillo, A. Gionis, and S. Leonardi, "Online team formation in social networks," in *Proceedings of the 21st international conference on World Wide Web*. ACM, 2012, pp. 839–848.

[10] A. Majumder, S. Datta, and K. Naidu, "Capacitated team formation problem on social networks," in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2012, pp. 1005–1013.

[11] M. Kargar, A. An, and M. Zihayat, "Efficient bi-objective team formation in social networks," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2012, pp. 483–498.

[12] M. Kargar, M. Zihayat, and A. An, "Finding affordable and collaborative teams from a network of experts," in *Proceedings of the 2013 SIAM International Conference on Data Mining*. SIAM, 2013, pp. 587–595.

[13] E. Fitzpatrick and R. Askin, "Forming effective worker teams with multi functional skill requirements," *Comput. Ind. Eng.*, vol. 48, no. 3, pp. 593–608, 2005.

[14] A. Baykasoglu, T. Dereli, and S. Das, "Project team selection using fuzzy optimization approach," *Cybernetics and Systems: An International Journal*, vol. 38, no. 2, pp. 155–185, 2007.

[15] H. Wi, S. Oh, J. Mun, and M. Jung, "A team formation model based on knowledge and collaboration," *Expert Syst. Appl.*, vol. 36, no. 5, pp. 9121–9134, 2009.

[16] M. Jackson, *Network formation.* The New Palgrave Dictionary of Economics and the Law, 2008.

[17] G. K. Awal and K. Bharadwaj, "Team formation in social networks based on collective intelligence–an evolutionary approach," *Applied Intelligence*, vol. 41, no. 2, pp. 627–648, 2014.

[18] H. Wi, S. Oh, J. Mun, and M. Jung, "A team formation model based on knowledge and collaboration," *Expert Systems with Applications*, vol. 36, no. 5, pp. 9121–9134, 2009.

[19] L. E. Blas, S. Sanz, E. G. Garcia, A. . Figueras, A. Bellido, and S. Fernandez, "Team Formation based on Group Technology: A Hybrid Grouping Genetic Algorithm Approach," *Comput Oper Res*, vol. 38, no. 2, p. 484–495, 2011.

[20] Z. C. Ani, A. Yasin, M. Z. Husin, and Z. A. Hamid, "A method for group formation using genetic algorithm," *International Journal on Computer Science and Engineering*, vol. 2, no. 9, pp. 3060–3064, 2010.

[21] F. Silva, C. Motta, F. Santoro, and C. Oliveira, "A Social Matching Approach to Support Team Configuration," in *CRIWG*, 2009, pp. 49–64.

[22] R. G. Reynolds, "An introduction to cultural algorithms," in *Proceedings of the third annual conference on evolutionary programming.* World Scientific, 1994, pp. 131–139.

[23] A. Engelbrecht, *Computational Intelligence: An Introduction.* John Wiley & Sons, 2007.

[24] E. Cohen, E. Halperin, H. Kaplan, and U. Zwick, "Reachability and distance queries via 2-hop labels," *SIAM Journal on Computing*, vol. 32, no. 5, pp. 1338–1355, 2003.

[25] T. Akiba, Y. Iwata, and Y. Yoshida, "Fast Exact Shortest-Path Distance Queries on Large Networks by Pruned Landmark Labeling," in *SIGMOD*, 2013, pp. 349–360.

Chapter 3

Team Formation in Community Based Palliative Care

Motivated by our first work in chapter 2, we examine the problem of imbalanced resource allocation and uneven service distribution in palliative care with the support of Hospice and the Windsor Essex Compassion Care Community in partnership with the Faculty of Nursing. This research proposes a model to address the above issues and provide an efficient computational solution.

3.1 Introduction

In recent years, it has been observed that geriatric populations are growing rapidly around the world. According to the UN [1], the number of individuals aged 60 years and over is projected to grow from 901 million in 2015 to 1.4 billion in 2030, which is a drastic 56 percent increase. In addition, it is projected that the global population of elderly people will reach 2.1 billion by 2050, accounting for approximately 20 percent of the world population.

Due to aging, the risk of chronic diseases, social isolation, depression, and fragmented care increases, along with other health related problems. This results in a

poorer self-perceived quality of life and an increased dependence on health care services for these individuals and their families. Consequently, the need of developing innovative solutions to support triple value healthcare (Personal, Technical and Allocative) must be considered as a critical issue for improving the quality and delivery of healthcare services [2, 3, 4]. The Personal value refers to the fact that, the decision making process is carried out based on the individual patient's values. Meanwhile, according to the Technical and Allocative values, the resources must be allocated and utilized optimally and equitably. It has been shown that improving health and overall well-being among elderly people can be achieved through: (1) enhancing their social support networks and (2) giving them a voice, and choices to make key decisions and direct their own care.

In recent years, person-centric and community-oriented palliative care systems are in the center of attention, to provide support for aging and other related challenges. Palliative care is a type of health care which focuses on improving the quality of life of individuals who are living with life-threatening illnesses, specially with chronic diseases such as cancer, cardiac disease, kidney failure, Alzheimer, ALS, and MS. The primary goal here is to provide various support services to help patients maintain an active life and dignity, while in some cases it may also positively influence the patients' prognosis and the course of illness. It also provides the patients' families with support, to help them better cope with the situation.

This care system uses a team approach to address the needs of patients and their families. In fact, a multidisciplinary team of healthcare professionals, volunteers, family members and friends work together to achieve a common goal of providing the optimal care services for a patient. This team forms a social circle of care for the patient. This paper propose a novel computational evolutionary model to form a team using members among a given palliative care network.

Generally, palliative care network include two groups of individuals; patients who

generally are not able to do some of their ordinary routine tasks, and care providers who are ready to offer a wide range of services to the patients, to cover their disabilities and support them with leading a normal life. However, each care provider has limited capabilities and can provide special type of services, while only having the capacity to support a limited number of patients. On the other hand, there are several barriers such as geographical distance, communication costs, time availability, etc., which make this process more complex.

In this study, approaching palliative care networks from a social point of view is suitable since it has a social structure. Consequently, this network will have a network of care providers and patients. Assuming that care providers are experts in providing a limited number of services and the patients need those services, making the social circle of care for each patient in an optimum manner can be seen as a team formation problem in social networks. In fact, based on the structure of the network, and relationship among the social actors in the network, the best team of support/care can be identified. Forming high-performance teams is very important, because the success of the care system is depend on their performance specially on how well the team members communicate and collaborate with each other and how quickly they can be available for offering the required services. Additionally, other factors such as their availability, geographical proximity and contact costs must also be considered for team formation. In real scenarios, taking these factors into account will lead to recommendation of a high performance team of care who can help a patient get back to leading a normal life.

In this research work, researchers assume that each patient has a profile which shows his/her capabilities. Capability here means the ability to do a task. The profile also determines the number and the type of the capabilities that a patient does not have, but is still required for a particular task. On the other hand, care providers also have a profile which represents the type and the number of services

that they can provide. Considering the distance, communication, and contact costs, the whole care network can be mapped to a weighted graph. Hence, the problem can be defined as identifying the best team of care (among all the care providers) who can support a patient by offering his/her required capabilities in the most cost effective manner. Moreover, at a system level, the challenge is to identify the optimal configuration of teams that will support as many patients as possible.

As the problem is an NP-hard problem, authors are proposing an evolutionary model based on the Cultural Algorithm (CA) to tackle it. CA is a knowledge-based evolutionary model which extracts different sources of knowledge from the best populations in each iteration, and uses them to guide the search direction to reach the near-optimal solution. In this research work, the primary steps are; keeping track of the best solution of each generation and extracting knowledge from them to form the situational knowledge. Then, this knowledge source are used to identify the suitable team of care. For the fitness function authors use a method proposed in [5, 6] to calculate the shortest path between the nodes.

The rest of the paper is organized as follows: In the next section, authors briefly review the related works. After that, in Section 3, our proposed model has been discussed and presented in detail. Evaluation and analysis of our model's performance is discussed and reviewed in Section 4, and finally it will conclude the whole idea of this paper.

3.2 Literature Review

Due to the increased importance of teamwork in the management and health care settings, researchers have started to examine the value of team formation [7]. However, only a small number of literature have focused on computational models for palliative health care systems. In this section, we briefly review some of the recent

research works categorized into three main related approaches: palliative care, team formation and evolutionary algorithms in health systems.

3.2.1 Palliative Care

As a novel approach in palliative care, an agent based model to improve the quality of service was proposed by the authors in [8]. They examined the method of assigning care providers in order to achieve the patient's goal. Based on contact costs and resource limitations, they worked on developing a framework for finding a group of suitable care providers to satisfy the requirements from patients. Their proposed model exhibited a reduction in operational costs and improvements in the quality of service [8].

The authors in [9] explored the challenge of balancing the physicians estimated prognoses with the actual care received versus the patients personal wishes. They addressed this using Deep Learning and Electronic Health Record (EHR) data to predict the all-cause mortality of a patient within the next 12 months, enabling the care team to take proactive measures towards reaching out to provide palliative care to such patients. The authors in [10] proposed an agent-based architecture to facilitate the communication and collaboration among the patients and care providers. Moreover, the authors of [11, 12] used both Multi-Agent Systems, and Information and Communication Technologies to improve the management of the clinical data of palliative care patients. In addition, authors of [13, 14, 15, 16] used multi-agent system to handle patient care system.

3.2.2 Team Formation

Social Network Team formation in health care networks is quite complex. The bulk of the literature on health care teams have focused on team functioning and performance. The authors in [17] proposed a method of forming a team of palliative care

network members, and emphasized the importance of collaboration among the members of the team. In addition, the authors in [18] explored the issues arising from challenges in communication among interdisciplinary palliative care team members. The authors in [19] analyzed how conflict develops among team members and explored several approaches to conflict resolution that might have been better used for caring or for comforting and loving the sick child. None of the models proposed were computational models. Therefore, various applications have been examined within research works which proposed a team formation problem using different approaches in social networks. As described in the previous section, palliative care networks can be considered as social networks. In the area of social networks of experts, much research has been conducted to date. The authors in [20, 21, 22] explored the team formation problem in the operational research community using branch-and-cut, genetic algorithms, and the Fuzzy inference approach respectively. However, the social structured graph among team members was not taken into consideration. Therefore, there was no survey or analysis of the effectiveness of collaboration among them. The authors in [23] explored the team formation problem by analyzing the connectivity between individuals in a social structured graph, and used the communication cost function to evaluate the effectiveness of the collaboration among them using enhanced Steiner algorithms. The authors in [24] generalized the team formation problem of [23, 24] and [25], and used a Density-based measure to find teams. The authors in [26] improved the cost function in [23] and introduced the concept of a leader in team formation and explored its significance. The authors in [27] explored this problem by only using work load balances between team members using an approximation algorithm. Then in [28] they examined the same problem with the addition of the communication cost factor. In reality, when individuals are employed on projects, a compensation is generally expected, with the exception of volunteering roles. The authors in [6] used personal cost combined with communication cost to form a team

of experts using an approximation algorithm. Later in [29], the team formation problem was examined with consideration for the expertise level of each member and their personnel cost. Furthermore, the authors in [30] introduced the concept of project profitability. Using a cluster hiring approach, they modeled the formation of the team of experts, hired to maximize profitability within a given budget. The authors in [31] examined a method to calculate the communication cost among the team, by using parameterized complexity analysis. The authors in [32] modeled teams as hierarchical structures to explore the ubiquitous nature of teams in real commercial and open source projects.

3.2.3 Evolutionary Algorithms in Health Systems

There is limited literature on the use of evolutionary algorithms on the team formation problem. The authors in [21] applied a genetic algorithm to choose team members and project managers, using a fuzzy inference system to calculate knowledge competence for the selection of the project manager. Additionally, the authors in [33, 34] also explored forming a team of experts using a genetic algorithm. In [35], the authors used a collective intelligent index to evaluate the expertness of each team member, and applied a genetic algorithm to find an optimal solution. The authors of [36] took a novel approach by considering the geographical location of each member of the team using a genetic algorithm. Furthermore, the authors in [5] applied a cultural algorithm to produce team of experts for various projects.

There is limited literature exploring the team formation problem in a health care setting. However, there is literature exploring the use of evolutionary algorithms to target various challenges in health care. In [37], the authors applied a genetic algorithms to handle the constraints related to workload balancing and multi-period planning. They also applied the principles of the robust optimization approach in a health care setting. The authors in [38] used genetic algorithms for health planning.

Based on the emergency cases and age related demographic factors, they found the optimal method of allocating ambulance services within a geographical locale.

In this paper, we use CA which is a knowledge-driven population-based evolutionary algorithms to optimize an efficient teams for patients in the palliative care networks. We optimize the communication cost, the distance cost and the contact cost of each team while we allocate certain number of tasks for each care providers. To the best of our knowledge, no prior work has been conducted to explore this problem using knowledge-based algorithm called CA as well as GA too. In this paper, we generated synthetic network using LHR benchmark with various number of nodes and allocate the nodes with higher number of degree as care providers and rest of them as patients. We compare the results from CA and GA with random method as a base algorithm. Although, there are limited works has been conducted in team formation in palliative care network, those are not computational models. The existing computational models which we discussed in above paragraphs used different strategy in their model.

3.3 Proposed Model

This section discuss our knowledge-based evolutionary model which is based on a cultural algorithm (CA), to find the best team of care providers for a patient.

3.3.1 Cultural Algorithm (CA)

CA is a dual inheritance evolutionary system consisting of population and belief spaces. During the optimization process, different sources of knowledge can be extracted from the best selected individuals of each iteration, which are then used to reduce the search domain and guide the search direction [39, 40].

Similar to other evolutionary algorithms such as genetic algorithms (GA), an

individual here means a possible solution for a given problem. Therefore, a population is a group of generated individuals in each iteration. First, the initial population is generated randomly. Then the performance of each individual is measured using a fitness function. Consequently, a group of individuals that have better relative fitness values are selected. In CA, different sources of knowledge (e.g., Normative, Situational, Historic, etc) will be extracted from this selected group. The assumption is that, understanding the knowledge behind their good performance can help us to generate a better population in subsequent iterations. Hence, a belief space is designed to store the extracted knowledge. In the next iteration, in addition to performing a crossover or a mutation, the CA uses the knowledge to guide the direction of the search and accelerate the evolution. The search process continues until the termination criteria is met. At the end of the process, the individual with the best fitness value is selected as the final solution for a given problem.

Algorithm 2 shows our proposed method for finding the best set of care provider teams for the patients in palliative care. The main components of our model are: representations, the fitness function, and the belief space structure, which will be reviewed in the next sections.

3.3.2 Representation

This research paper considers three main entities (patient, care provider, and individual) which must be defined formally in advance. Similar to the model proposed in [8], each patient is represented by a binary array with a fixed size of m, where m is the number of capabilities. Each cell of the array indicates a predefined capability c, where the value is 1 if the patient has that capability, and 0 otherwise. For example, $P_1 = [0, 1, 1, 1, 1, 0]$ represents a patient with the index id of 1, who requires the two capabilities c_0 and c_5, where c_0 may represent the ability to drive, and c_5 can represent the ability to walk.

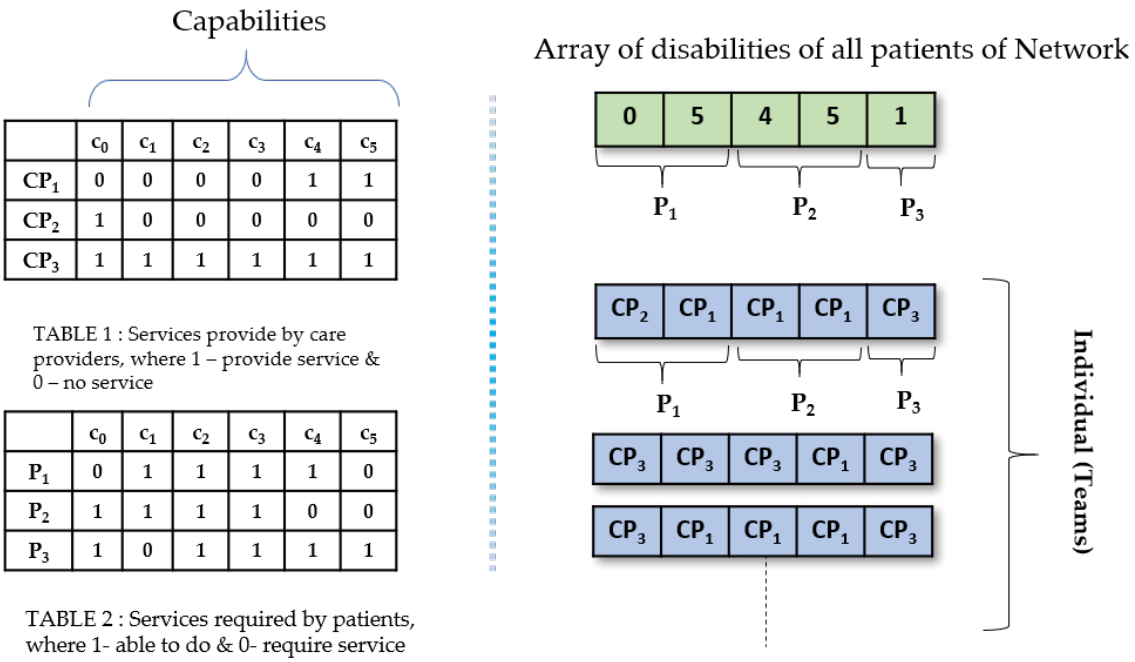
Algorithm 2 Team formation in Palliative Care

Input: graph G as a network of patient and care providers; RC as a set of required capabilities of patients; list of available capabilities provided by care providers (CP)
Output: teams of care providers for all patients$solution$

```
1: n ← |population|
2: gs ← number of iterations
3: t ← |selected population|
4: elite ← number of elite individuals
5: Pop_(1...n) ← generate random individuals
6: for i ← 1 gs do
7:    Fit_Pop ← Fitness(Pop)
8:    (Pop) ← Sort_(Fit)Pop
9:    SP ← Pop_(1...t)
10:   BS ← SP^T
11:   for j ← elite n do
12:      if random() ≤ 80% then
13:         Pop_j ← generateIndividual(BS)
14:      else
15:         if random() ≤ 80% then
16:            Pop_j ← Crossover
17:         else
18:            Pop_j ← Mutation
19:         end if
20:      end if
21:   end for
22: end for
23: solution ← Pop_1
```

A care provider also is represented by a fixed-size binary array similar to the patient. The only difference is that, if a value in a cell is 1, it means that the care provider can provide that capability. In addition, each care provider has a maximum capacity for providing services, which has to be set in advance. For example, each care provider can provide a service to a maximum of 5 patients.

Finally, to represent an individual (a candidate solution), we use an array structure with n cells, where s is the length of the array is equal to the total number of required capabilities. Let $RC = \{RC^{P_1}, RC^{P_2}, ..., RC^{P_n}\}$ represent a set of required capabilities for all patients (P), where n is the total number of patients in the network. Then, $RC^{P_i} = \{(rc_j, ..., rc_m) | rc_j \in \{C^{P_i} \neq 1\}\}$. For example, as shown in Figure. 3.1,



	c_0	c_1	c_2	c_3	c_4	c_5
CP_1	0	0	0	0	1	1
CP_2	1	0	0	0	0	0
CP_3	1	1	1	1	1	1

TABLE 1 : Services provide by care providers, where 1 – provide service & 0 – no service

	c_0	c_1	c_2	c_3	c_4	c_5
P_1	0	1	1	1	1	0
P_2	1	1	1	1	0	0
P_3	1	0	1	1	1	1

TABLE 2 : Services required by patients, where 1- able to do & 0- require service

Figure 3.1: Sample random individuals: the representation of teams for the whole patients of a palliative care networks which have 3 care providers and 3 patients.

if there are three patients in the network and the total number of required capabilities is five, the size of the individual array, s, is equal to five. The first two values are the required capabilities of patient 1 (c_0&c_5), the next two are the needed capabilities of patient 2 (c_4&c_5) and the last cell is the required capability of patient 3 (c_1). The value of each cell is the index of a care provider selected from the set of available care providers who possess those capabilities. In fact, these sets are generated in the initialization phase, hence for each capability, a pool of care providers that can provide that capabilities are created. In Figure. 3.1, we have three care providers, CP_1, CP_2, and CP_3. Care providers 2 and 3 can offer the capability of c_0, while care provider 3, is the only provider who offers capabilities of c_1, c_2 and c_3. In addition, capabilities of c_4 and c_5 are also provided by care providers 1 and 3. According to the previous example shown in Figure. 3.1, the first patient needs the two capabilities of c_0 and c_5, the next patient requires the two capabilities of c_4 and c_5 and the last patient needs just the capability of c_1. Consequently, three random individuals

or potential solutions for this problem are illustrated in Figure. 3.1. According to $Individual_i$ which is generated randomly, CP_2 and CP_1 form the care team for patient 1. CP_1 is also responsible for supporting patient 2, and care provider 3 is assigned to assist patient 3. As shown in this example, various types of teams can be generated using this individual representation method. However, not all of the teams generated represent the optimal solution. Consequently, we have to evaluate the performance of the solutions using the fitness function.

3.3.3 Fitness Function

As mentioned previously, our model uses a weighted graph to show the relationship among the social entities. The weight here is calculated based on the three groups of costs. The first is the communication cost, denoted by $Ccost$, which represents how easily each pair of people can interact with each other. The value is between 0 and 1, with a lower value indicating a better level of communication between a pair of people. Another group is the distance cost, denoted by $Dcost$, which refers to the geographical distance between the social actors. Similar to the previous cost, the value is between 0 and 1 and a lower value indicates an increased chance of cooperation. The last one is contact cost, denoted by $Tcost$ which shows the level of productivity of a social entity. The value is again between 0 and 1 but our goal is to maximize this value.

To measure the performance of our algorithm, authors are using the following fitness function which is based on the formula proposed in [6]:

$$F(I) = \sum_{i=1}^{n} F1(TP_i) \tag{3.1}$$

where TP_i is a generated team for the patient i.

$$F1(TP) = \lambda\, CommunCost + \beta\, DistCost + \gamma\, (1 - ContactCost). \tag{3.2}$$

where λ, β, and γ are balance factors.

Given a team TP of care providers for a patient: $\{(c_1, CP_1), (c_2, CP_2), ..., (c_m, CP_k)\}$, the sum of distances of TP with respect to $Ccost$, $Dcost$ and $Tcost$ among the pair of social actors is defined as,

$$CommunCost = \sum_{i=1}^{n} \sum_{j=i+1}^{n} Ccost_i, j \tag{3.3}$$

$$DistCost = \sum_{i=1}^{n} \sum_{j=i+1}^{n} Dcost_i, j \tag{3.4}$$

$$ContactCost = \sum_{i=1}^{n} Tcost_i \tag{3.5}$$

where i and j are indexes of a pair of social actors in the network.

3.3.4 Belief Space

Our approach to make the belief space, which is a knowledge-based repository, has been inspired from the belief space formation model proposed in [41]. It is defined as the transpose matrix of the selected individuals. Let a selected individual be defined as $SI_i = [TP_i, TP_2, \ldots, TP_n]$. Now, assuming the number of the selected individuals in each iteration is t , the selected population can be defined as follows:

$$SP = \begin{bmatrix} SI_1 \\ SI_2 \\ \vdots \\ SI_t \end{bmatrix} = \begin{bmatrix} TP_1^1, TP_2^1, \ldots, TP_n^1 \\ TP_1^2, TP_2^2, \ldots, TP_n^2 \\ \vdots \\ TP_1^t, TP_2^t, \ldots, TP_n^t \end{bmatrix}$$

Thus, the belief space is defined as $BS = SP^T$:

$$BS = \begin{bmatrix} TP_1^1, TP_1^2, \ldots, TP_1^t \\ TP_2^1, TP_2^2, \ldots, TP_2^t \\ \vdots \\ TP_n^1, TP_n^2, \ldots, TP_n^t \end{bmatrix}$$

In other words, for each capability (a row of the BS matrix), the BS matrix contains the list of care providers who have previously appeared among the selected individuals.

Assuming the optimal solution can be generated by using the extracted knowledge from the best individuals, in the subsequent iteration, the algorithm generates a new set of individuals by reading the data from the BS matrix and not the pool of care providers. As a result, we expect to observe a reduction in the size of the search space in each subsequent iteration.

3.4 Evaluation

This section reports the performance evaluation of our model.

We have taken into consideration 4 different synthetic social networks (i.e. 25, 50, 75, 100 nodes), by grouping patients and care providers using various ratios such as the following:

1. 25 percentage of patients to 75 percentage of care providers, where care providers can provide a maximum of 3 services at a time

2. 30 percentage of patients to 70 percentage of care providers, where care providers can provide a maximum of 4 services at a time

3. 50 percentage of patients to 50 percentage of care providers, where care providers

can provide a maximum of 6 services at a time

The networks are generated based on LFR benchmark for generating social networks[42] with the default setting. In addition, Communication, Distance and Contact cost have been assigned to the network randomly.

The random approach has been used as a base model for comparison. The random approach involves randomly assembling a team of care providers. The fitness value obtained from random approach is used to compare against the fitness values obtained from our Cultural Algorithm, and a Genetic algorithm. We tested our model on a system with the following specification:

1. Installed memory (RAM): 16 GB,
2. Processor: Intel® Core™ i5 CPU @ 2.50 GHz.
3. Java was used to develop the experimental model.

Each experiment has been conducted 5 times independently, to find the average fitness values. The fitness values has been calculated using the fitness function based on the weighted importance of the cost parameters $\alpha, \beta, and \gamma$, which were assigned the values of 0.6, 0.3, and 0.1 respectively.

Figures 3.2, 3.3 and 3.4 exhibit the comparison between the fitness value against different network sizes, tested using various algorithms such as a Cultural and Genetic algorithm, and also using a random approach. As shown in Figure. 3.2, when 25% of the population are patients, and the rest are care providers, our algorithm can find a near optimal team of care providers with the fitness values of 14.12, 16.61, 20.33, 29.03, when the size of the network ranges between 25 to 100 nodes. The fitness values obtained using the genetic algorithm and through the random approach perform relatively weaker than our proposed algorithm.

Figure. 3.3 represents the results obtained for a network consisting of 30% of patients. We can observe the similar patterns between our approach and the other

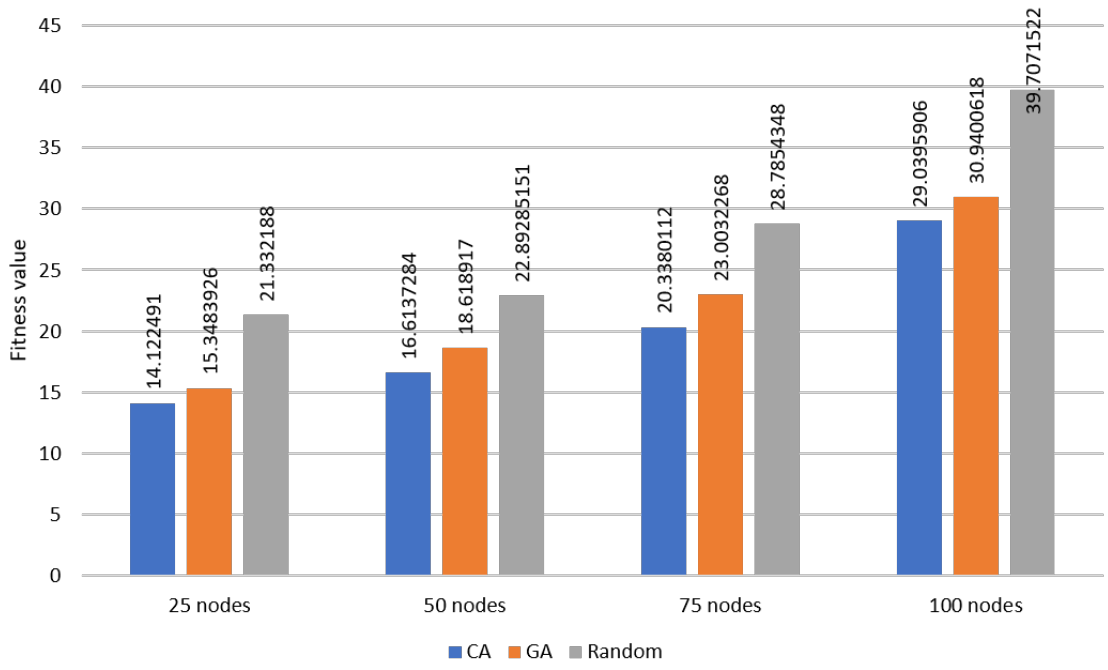


Figure 3.2: Experimental results for 25% patients

methods. For example, when the fitness value of our model is 31.44 for a network with 100 nodes, the value is 35.85 and 47.38 for the genetic and random approaches respectively. This exhibits approximately a 13% higher performance relative to the genetic approach, and 34% relative to the random approach.

As illustrated in Figure. 3.4, when the proportion of patients and care providers are equal, the performance of the algorithms are relatively similar, as expected. This can be attributed to the fact that we need to have enough care providers to meet the needs of every patient. However, even in this scenario, our algorithm outperforms the other algorithms.

According to the results, our proposed model and algorithm showed an overall better performance in comparison to the other algorithms tested to find a near optimal team of care providers for the patients in a community-based palliative care network.

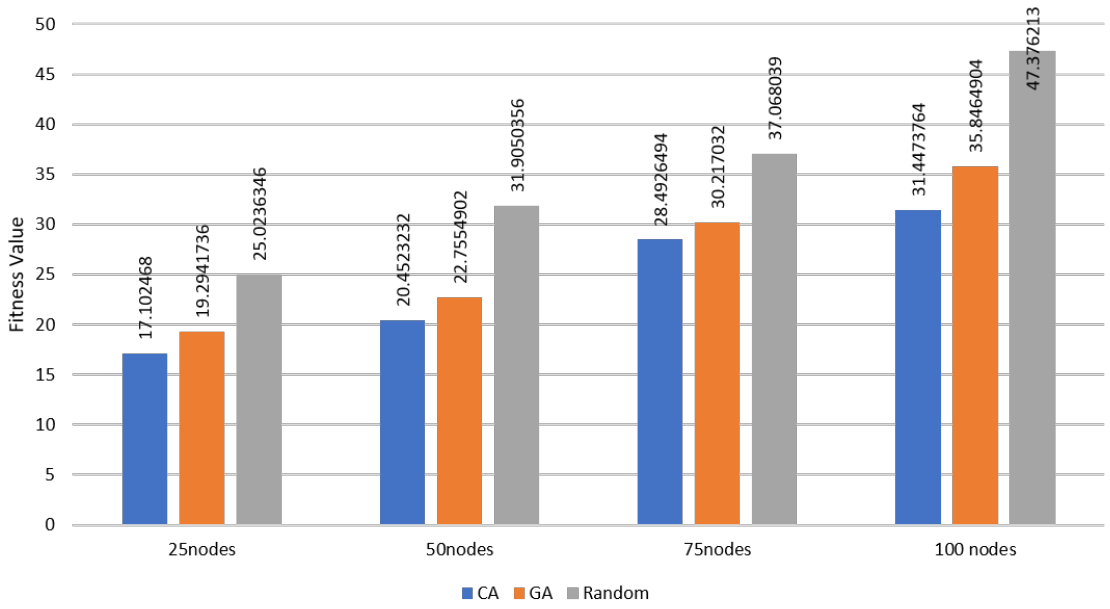


Figure 3.3: Experimental results for 30% patients

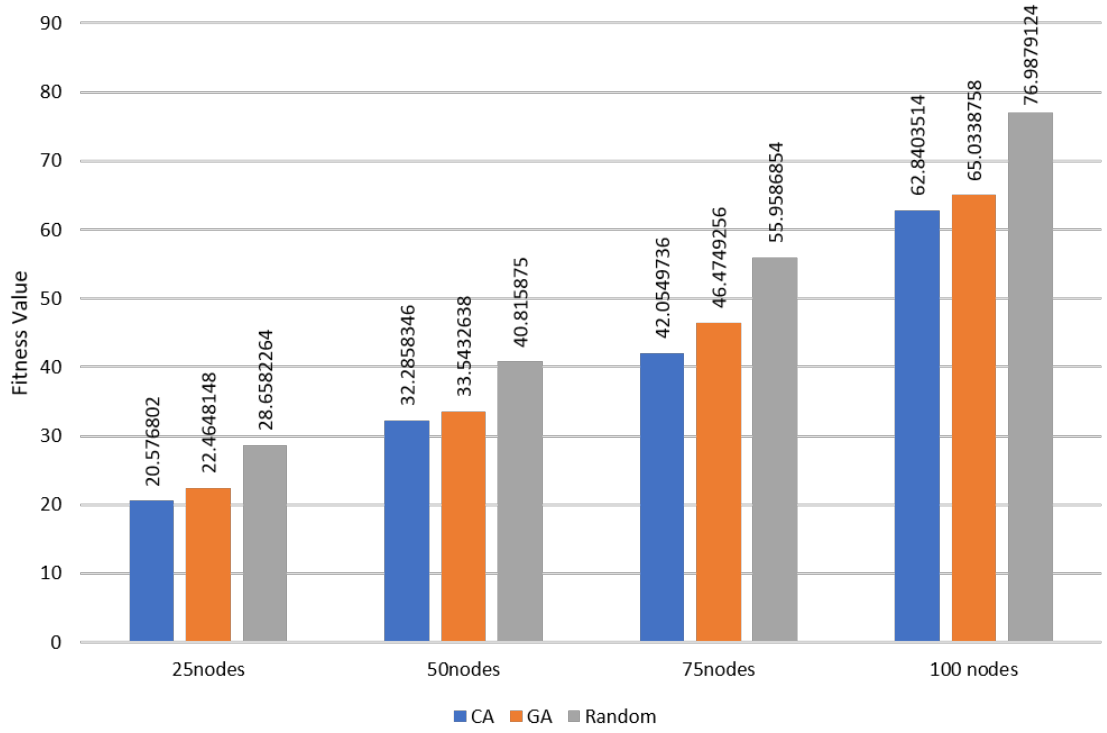


Figure 3.4: Experimental results for 50% patients

3.5 Conclusion

In this paper, we proposed a novel approach of assembling a team of care providers for palliative care patients in a community-oriented setting. Our model consists of two

primary social entities, patient and care providers, who interact with each other in a social network context. The patients require some support capabilities to lead a normal life style and the care providers offers these capabilities demanded by the patients. Authors took into a consideration three different cost variables, communication, contact, and distance costs. The overall goal of our research was to minimize the costs and maximize patient satisfaction. A model has been developed using a knowledge-based evolutionary algorithm to optimize the resource allocation and team generation processes in order to provide patients with added value in the form of quality service delivery and an increased quality of life. The results obtained from our evaluation indicate that, our model is more effective at obtaining the near optimal team formation solution relative to the other algorithms currently proposed in literature within the field.

In future we are going to enhance the algorithm and validate its performance against clinical data. In addition, we plan to expand the experimental scope and take additional parameters into considerations.

Bibliography

[1] J. Melorose, R. Perroy, and S. Careas, "World population prospects: The 2015 revision, key findings and advance tables," in *Working Paper No. ESA/P/WP. 241*, 2015, pp. 1–59.

[2] A. Jani, S. Jungmann, and M. Gray, "Shifting to triple value healthcare: Reflections from england," *Zeitschrift für Evidenz, Fortbildung und Qualität im Gesundheitswesen*, 2018.

[3] M. Gray, G. Wells, and T. Lagerberg, "Optimising allocative value for populations," *Journal of the Royal Society of Medicine*, vol. 110, no. 4, pp. 138–143, 2017.

[4] A. Silenzi and S. Boccia, "Value based healthcare as a solution for the future of publicly funded healthcare systems," *Epidemiology, Biostatistics and Public Health*, vol. 14, no. 4, 2017.

[5] K. Selvarajah, P. M. Zadeh, M. Kargar, and Z. Kobti, "A knowledge-based computational algorithm for discovering a team of experts in social networks," 2017.

[6] M. Kargar, A. An, and M. Zihayat, "Efficient bi-objective team formation in social networks," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2012, pp. 483–498.

[7] L. Kalra, A. Evans, I. Perez, M. Knapp, N. Donaldson, and C. G. Swift, "Alternative strategies for stroke care: a prospective randomised controlled trial," *The Lancet*, vol. 356, no. 9233, pp. 894–899, 2000.

[8] N. Moradianzadeh, P. M. Zadeh, Z. Kobti, and K. Pfaff, "An agent model to support social network-based palliative care," in *Computers and Communications (ISCC), 2017 IEEE Symposium on.* IEEE, 2017, pp. 300–305.

[9] A. Avati, K. Jung, S. Harman, L. Downing, A. Ng, and N. H. Shah, "Improving palliative care with deep learning," *arXiv preprint arXiv:1711.06402*, 2017.

[10] J. Ruan, W. MacCaull, and H. Jewers, "Enhancing patient-centered palliative care with collaborative agents," in *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2010 IEEE/WIC/ACM International Conference on*, vol. 3. IEEE, 2010, pp. 356–360.

[11] A. Moreno, D. Riano, and A. Valls, "Agent-based alarm management in a palliative care unit," in *III Workshop on Agents Applied in Health Care, 19th International Joint Conference on Artificial Intelligence (2005)*, 2005, pp. 6–60.

[12] D. Riano, A. Moreno, and A. Valls, "Palliasys: Agent-based palliative care," in *IEEE 4th Conf on Intelligent Systems Design and Applications ISDA*, vol. 4, no. 2004, 2004, pp. 1–6.

[13] D. Riaño, S. Prado, A. Pascual, and S. Martín, "A multi-agent system model to support palliative care units," in *Computer-Based Medical Systems, 2002.(CBMS 2002). Proceedings of the 15th IEEE Symposium on.* IEEE, 2002, pp. 35–40.

[14] D. Isern, M. Millan, A. Moreno, G. Pedone, and L. Z. Varga, "Home care individual intervention plans in the k4care platform," in *Computer-Based Medical Systems, 2008. CBMS'08. 21st IEEE International Symposium on.* IEEE, 2008, pp. 455–457.

[15] C. Aouatef, B. Iman, and C. Allaoua, "Multi-agent system in ambient environment for assistance of elderly sick peoples," in *Proceedings of the International Conference on Intelligent Information Processing, Security and Advanced Communication*. ACM, 2015, p. 21.

[16] A. J. Singh, P. Dalapati, and A. Dutta, "Multi agent based dynamic task allocation," in *Agent and Multi-Agent Systems: Technologies and Applications*. Springer, 2014, pp. 171–182.

[17] O. Spruyt, "Team networking in palliative care," *Indian journal of palliative care*, vol. 17, no. Suppl, p. S17, 2011.

[18] A. Street and J. Blackford, "Communication issues for the interdisciplinary community palliative care team," *Journal of clinical nursing*, vol. 10, no. 5, pp. 643–650, 2001.

[19] A. Verhagen, "Teamwork and conflicts in paediatric end-of-life care," *Acta Paediatrica*, vol. 107, no. 2, pp. 192–193, 2018.

[20] A. Zzkarian and A. Kusiak, "Forming teams: an analytical approach," *IIE transactions*, vol. 31, no. 1, pp. 85–97, 1999.

[21] H. Wi, S. Oh, J. Mun, and M. Jung, "A team formation model based on knowledge and collaboration," *Expert Systems with Applications*, vol. 36, no. 5, pp. 9121–9134, 2009.

[22] A. Baykasoglu, T. Dereli, and S. Das, "Project team selection using fuzzy optimization approach," *Cybernetics and Systems: An International Journal*, vol. 38, no. 2, pp. 155–185, 2007.

[23] T. Lappas, K. Liu, and E. Terzi, "Finding a team of experts in social networks,"

in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2009, pp. 467–476.

[24] C.-T. Li and M.-K. Shan, "Team formation for generalized tasks in expertise social networks," in *Social Computing (SocialCom), 2010 IEEE Second International Conference on*. IEEE, 2010, pp. 9–16.

[25] A. Gajewar and A. Das Sarma, "Multi-skill collaborative teams based on densest subgraphs," in *Proceedings of the 2012 SIAM International Conference on Data Mining*. SIAM, 2012, pp. 165–176.

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

[27] A. Anagnostopoulos, L. Becchetti, C. Castillo, A. Gionis, and S. Leonardi, "Power in unity: forming teams in large-scale community systems," in *Proceedings of the 19th ACM international conference on Information and knowledge management*. ACM, 2010, pp. 599–608.

[28] ——, "Online team formation in social networks," in *Proceedings of the 21st international conference on World Wide Web*. ACM, 2012, pp. 839–848.

[29] M. Zihayat, M. Kargar, and A. An, "Two-phase pareto set discovery for team formation in social networks," in *Web Intelligence (WI) and Intelligent Agent Technologies (IAT), 2014 IEEE/WIC/ACM International Joint Conferences on*, vol. 2. IEEE, 2014, pp. 304–311.

[30] B. Golshan, T. Lappas, and E. Terzi, "Profit-maximizing cluster hires," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2014, pp. 1196–1205.

[31] R. Bredereck, J. Chen, F. Hüffner, and S. Kratsch, "Parameterized complexity of team formation in social networks," *Theoretical Computer Science*, 2017.

[32] C. Ding, F. Xia, G. Gopalakrishnan, W. Qian, and A. Zhou, "Teamgen: An interactive team formation system based on professional social network," in *Proceedings of the 26th International Conference on World Wide Web Companion*. International World Wide Web Conferences Steering Committee, 2017, pp. 195–199.

[33] Z. C. Ani, A. Yasin, M. Z. Husin, and Z. A. Hamid, "A method for group formation using genetic algorithm," *International Journal on Computer Science and Engineering*, vol. 2, no. 9, pp. 3060–3064, 2010.

[34] L. E. Agustín-Blas, S. Salcedo-Sanz, E. G. Ortiz-García, A. Portilla-Figueras, Á. M. Pérez-Bellido, and S. Jiménez-Fernández, "Team formation based on group technology: A hybrid grouping genetic algorithm approach," *Computers & Operations Research*, vol. 38, no. 2, pp. 484–495, 2011.

[35] G. K. Awal and K. Bharadwaj, "Team formation in social networks based on collective intelligence–an evolutionary approach," *Applied Intelligence*, vol. 41, no. 2, pp. 627–648, 2014.

[36] Y. Han, Y. Wan, L. Chen, G. Xu, and J. Wu, "Exploiting geographical location for team formation in social coding sites," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2017, pp. 499–510.

[37] T. V. L. Nguyen and R. Montemanni, "Integrated home health care optimization via genetic algorithms and mathematical programming," in *Evolutionary Computation (CEC), 2016 IEEE Congress on*. IEEE, 2016, pp. 553–561.

[38] S. Sasaki, A. J. Comber, H. Suzuki, and C. Brunsdon, "Using genetic algorithms

to optimise current and future health planning-the example of ambulance locations," *International journal of health geographics*, vol. 9, no. 1, p. 4, 2010.

[39] R. G. Reynolds, "An introduction to cultural algorithms," in *Proceedings of the third annual conference on evolutionary programming*. World Scientific, 1994, pp. 131–139.

[40] A. Engelbrecht, *Computational Intelligence: An Introduction*. John Wiley & Sons, 2007.

[41] P. M. Zadeh and Z. Kobti, "A Multi-population Cultural Algorithm for Community Detection in Social Networks," *Procedia Computer Science*, vol. 52, no. 9, pp. 342–349, 2015.

[42] A. Lancichinetti, S. Fortunato, and F. Radicchi, "Benchmark graphs for testing community detection algorithms," *Physical review E*, vol. 78, no. 4, p. 046110, 2008.

Chapter 4

A Palliative Care Simulator and Visualization Framework

Chapter 3 describes the problems and the computational solution for the palliative care system. However, we believe that rather than just showing results as numerical format, visualizing the networks and team distribution to each patient are flexible for anyone who uses the framework. This chapter explains how we develop a visualization framework for any health care network to see the solution and handle real data.

4.1 Introduction

Palliative care is a special type of care that aims to enhance the quality of life of patients and families who are dealing with life-threatening illnesses. A key objective here is to help patients maintain an active life and dignity by providing them a diverse range of support services.

Generally, palliative care is a team-oriented care system where multidisciplinary teams of formal and informal care providers including healthcare professionals, volunteers, family members, and friends, work together to support the patients [1, 2]. Consequently, examining the nature of relationships and interactions among the team

members can be useful for optimizing the overall performance of the care system and improving the efficiency of the teams.

Our approach to model palliative care is to convert it to a social network graph. As a result, we can apply social network analysis techniques to identify the underlying structures of the network and its evolution and formation. Social network graphs usually made up of two elements: individuals (nodes) and the social ties (edges) between them. Consequently, in a palliative care network, patients and care providers can be considered as the nodes and the relationships between them as edges. Patients lack some basic capabilities which prevent them to have a high quality life-style. On the other hand, care providers are ready to support and assist the patients to cover their shortcomings. Across the cities, many hospices connect patients and care providers and try to optimize the patient-care providers ratio. Although, there have been major improvement in the care services during the last decade,unequal distribution of service accessibility is still a big challenge in this field [2].

On the other hand, data visualization is a powerful tool which gives us a clearer understanding of the structure and behaviour of a given system and its components, either by measurement or providing visual insight [3]. It is a robust methodology for analyzing the complex network structures. The focus of this paper is on the visualization of palliative care networks. The schematic view of our model and designed tools are shown in Figure. 4.1. To the best of our knowledge, this is the first framework to visualize and analyze palliative care networks.

This framework allows the patients and care providers to explore and monitor the evolution and changes in the cohesion and structure of the care network. It is capable of analyzing real data as well as synthetic data and it can be used to identify the isolated patients, imbalance resource allocation, and uneven service distribution in the network.

The rest of the paper is organized as follows: The next section briefly reviews

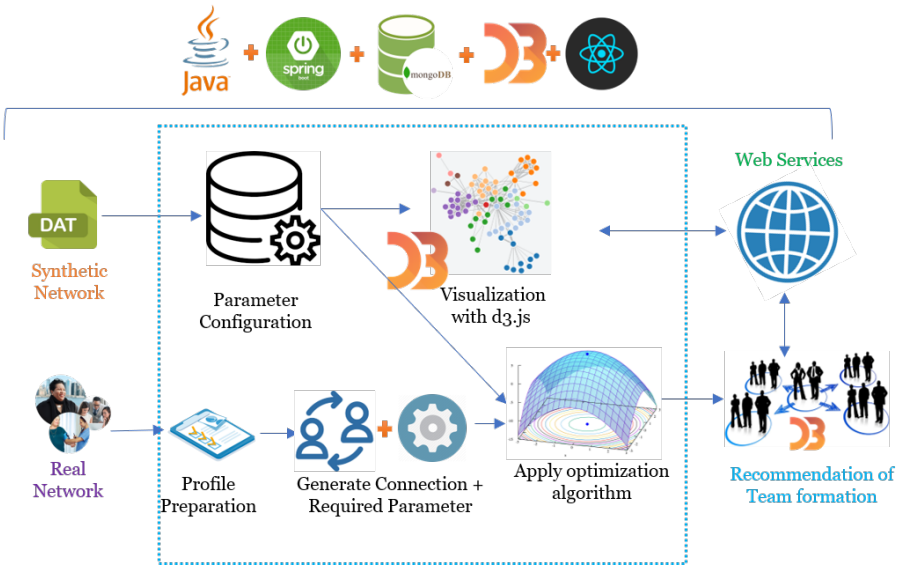


Figure 4.1: The schematic view of the proposed model.

the related work. In section 3, we present our proposed visualization framework. In section 4, we discuss the experimental setup. Finally, section 5 concludes the idea of this paper with directions for future work.

4.2 Related Work

Data visualization has a long and strong history in science, particularly in network-based systems. It plays a vital role in various types of social network studies including crime analysis, and sociological, organizational and epidemic studies. In this section we briefly review some the recent works in the field of healthcare and social network visualization.

In [4], the authors proposed a new approach to use visualization techniques in social network analysis in order to enhance the performance of the analysis by incorporating statistical measures. The authors in [5], highlighted challenges and opportunities of big data visualization and analysis on social networks. They also proposed a new method for visualizing the big data. In [6], the authors designed an interactive tool to visualize the influence networks of artists. The authors in [7], proposed

a novel technique to visualize a network using a hierarchical structure. In [8], the authors proposed a visualization tool for monitoring a health network and its status with focusing on the scalability issue.

On the other hand, the importance of teamwork in palliative care has been addressed in few computer science research works. The authors in [9], used Electronic Health Record (EHR) data to create a system to prioritize the palliative care patients for the follow-up meetings. The authors in [10], have proposed a new agent-based model to allocate care providers to patients in order to maximize the satisfaction rate and reduce the operational costs. The authors in [2], modeled palliative care using a team-based approach.

However, still there is not any significant work and framework for visualization of palliative healthcare system. This paper will discuss the dynamic way to analysis the palliative care networks.

4.3 A Framework for Palliative Care Visualization

As mentioned before, our proposed framework is capable of visualizing the structure of the palliative care network. In addition, it can be used for identifying and visualizing suitable teams of care for a group of patients. This feature allows the care providers and administrators to identify the best teams of experts for any given care network. It also can be used to identify the isolated patients and imbalance service allocation in the system. In this section we describe different components of this framework. As shown in Figure. 4.1, our framework generally consists of four main components which are Data Entry, Computational Engine, Visualization, and Dynamic network representation. These components are linked together to provide a series of descriptive and predictive analytic tools for the care providers and policy-makers.

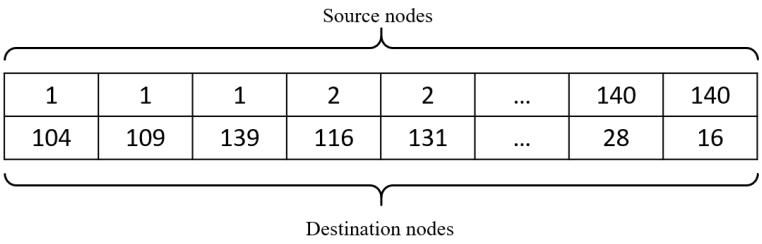


Figure 4.2: An example of a synthetic network data with 140 nodes.

4.3.1 Data Entry Unit

The Data Entry Unit is responsible to receive the care's network data from the user, store and pre-process it for the further analysis. The input data can be synthetic or real data. In the case of synthetic network, our model use the LFR benchmark[1] [11] which is an internationally recognized social network generator benchmark to create a synthetic network. The format of the generated network can be seen as an array of n nodes, where n is the size of the network. For example in Figure. 4.2, we have a network with 140 nodes and each column shows a link between a source and destination node. Users can generate various networks using this benchmark in different sizes and complexities. However, the generated synthetic network only represents the structure of the network. In other words, it just shows who is connecting to whom in the network, but it does not determine if the node is a patient or a care provider. Consequently, after generating the synthetic network, this unit provides a rich Graphical User Interface (GUI) to assign roles to each node, as well as the list of capabilities of each patient and the list of services that each care providers can handle. These features can be assigned automatically by the implemented algorithm or manually by the client. By assigning these features, the synthetic network can be seen as a real network and will be ready for the further analysis.

In the case of real-world data, actual profiles of the members of a given palliative care network are uploaded to the system and the unit is responsible to create the

[1]https://sites.google.com/site/andrealancichinetti/files

social graph based on that. As the framework supports multiple views, it can be used by either patients or care providers. Using the GUI (Graphical User Interface), both groups can pre-process the data by editing or adjusting the missing values or fixing the anomalies in the graph.

After that, the processed data will be stored in the system in a NOSQL database and will be shared between other components for analysis and visualization.

4.3.2 Computational Engine Unit

This unit provides a set of social network analysis and machine learning techniques to process a given network. First, a community detection algorithm is applied on the network to identify the clusters and their memberships. As a result of this step, the network will be divided into multiple communities and the outliers and socially isolated nodes will be determined. It can help the policy-makers and care providers to understand the underlying structure of the network and find some patterns and similarity indexes among the community members. It also can identify highly influential members in the network.

In addition, the clustering process is used to identify imbalanced service allocation in the care network. For example, if there is a nurse or a care centre in the network that provide services to a large number of patients and at the same time another centre has a very limited number of patients, the algorithm identify them and marks them for the further process and optimization. For the community detection algorithm, our model uses the existing algorithm proposed in [12]. This is a knowledge-based clustering algorithm that uses a variation of Cultural algorithms to identify the communities on a given social network. The knowledge extracted from this process is used during the next analytic steps.

Another important process which is done in this unit is Team formation which can be defined as a process of allocating suitable experts to complete a specific task. As

we mentioned before, the palliative care is a highly team-oriented health care process. Due to its complex nature, generally a wide range of community services is required for a patient to have a normal life. Assuming a big list of formal and informal care providers, finding a team of care that can work together efficiently in an optimal way to cover all the patient's needs is a team formation problem.

The authors in [2] proposed a method for team formation in palliative care networks. In their work a cultural algorithm [13] has been proposed to optimize the task of allocating teams of care providers to the entire patients in a given network and maximizing their satisfaction level. The optimal teams of experts can be formed in order to satisfy patients' requirements and other parameters such as communication cost, geographical proximity, availability, and workload. Consequently, for the team formation, we base our algorithm on the work published in [2].

4.3.3 Data Visualization Unit

This unit is responsible to visualize the processed data. Generally, analyzing trends and patterns in large data sets is a very complex process, and data visualization is a very useful technique to simplify this process and it enables decision-makers to derive analytical results from the visually presented information. Consequently, visualization of the processed data obtained from a given healthcare network can be useful to understand and monitor the evolution and the hidden patterns of the network. In addition, improving the care services, and optimizing the connections between people all can be achieved using this visualization module. This unit is designed to represent the raw structure of the network as a social graph, clusters and communities, circle of care of each patients, the level of distance and similarity between each two nodes in the graph.

4.3.4 Dynamic Network Representation

This unit is responsible to represent the evolution of the network in a period of time. This time-based representations helps the policy-maker to travel in time and identify patterns and trends in the network. It also can be used as a predictive model to predict the future state of the network based on the historical data obtained by the system.

4.4 Experimental Analysis and Implementation

In this section we discuss the implementation details of our proposed framework. The visualization of the palliative care networks was mainly generated by d3.js. In addition, we used Java and Spring Boot framework for our back-end works and MongoDB was used to manage the database for real network setting. Our front-end has been implemented with React and Bootstap.

As discussed earlier, the user can decide what type of datasets they are going to explore, real or synthetic data. Figure. 4.3 shows the interface for uploading the synthetic dataset.

To evaluate the performance of the framework, we have populated that with a synthetic network generated by LFR benchmark. After generating a synthetic social care network, we have to assign roles and other features to the network. Figure.4.5 shows the UI for assigning the required values to the network in order to imitate the real data. At the first, some basic structural information about a given social graph (e.g. the number of nodes and edges, the average degree of a node) is automatically calculated and presented to the user. After that, the user can generate a customized care network using the following parameters:

i Ratio of patients to care providers: The user can determine how many of the

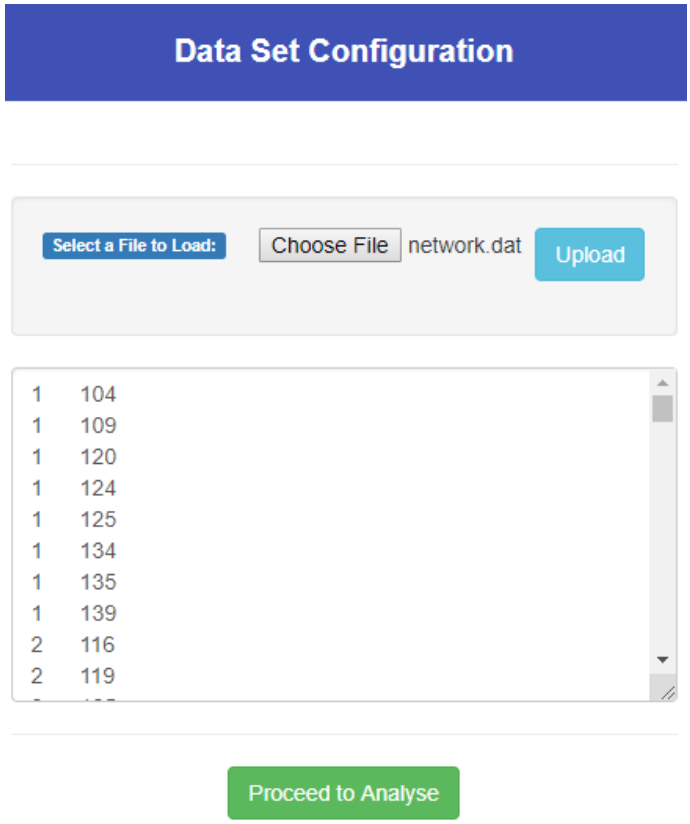


Figure 4.3: The interface for uploading synthetic dataset

nodes in the network are patients and how many are care providers by assigin a ratio in this field.

ii Distribution: The user can choose the way that care providers and patients are labeled in the network. The Framework provides three methods for this feature which are ordered, random or betweenness centrality.

The ordered means that the nodes are arranged based on their degree of connections, then the nodes with higher degrees are marked as care providers and the rest will be patients. The random distribution assigns labels randomly to the nodes, in order to meet a given number of patients and care providers. The betweenness centrality indicates the number of times that a node acts as a bridge along the shortest path between two other nodes. The nodes which have a high betweenness centrality measure consider as care providers and remaining will be

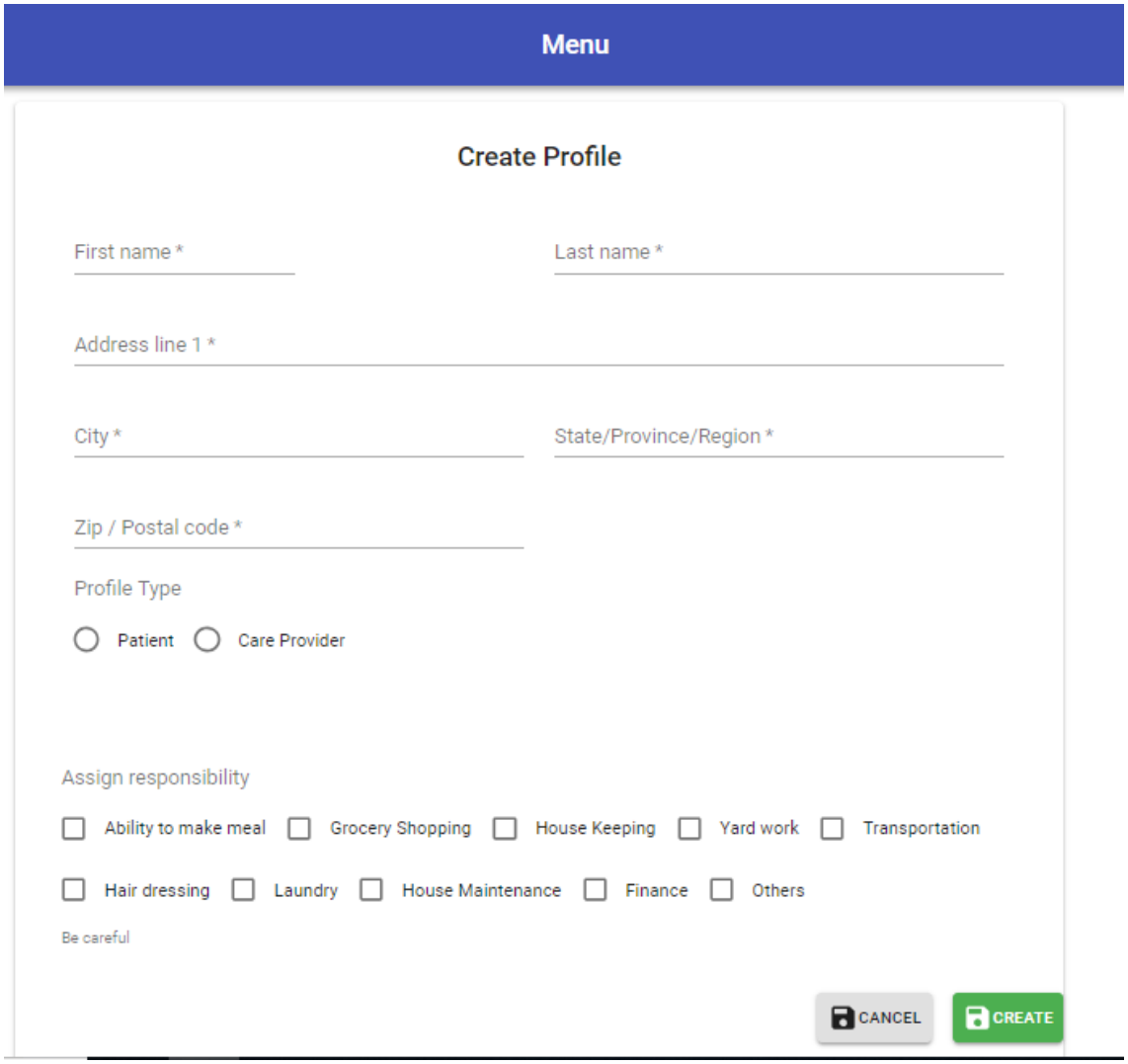


Figure 4.4: The interface for uploading profiles of Real dataset

marked as patients.

iii Number of Capabilities: the total number of capabilities that is going to be considered for a given network. In the palliative care, various types of services or capabilities are needed by a patient, for example a patient may not be able to make its own Meals, or doing grocery shopping, House Keeping, and laundry. For making the synthetic care network, the user can define the number of capabilities that must be considered for patients in the network. It can be either filled with the names of those capabilities (e.g. grocery shopping) or be filled without pro-

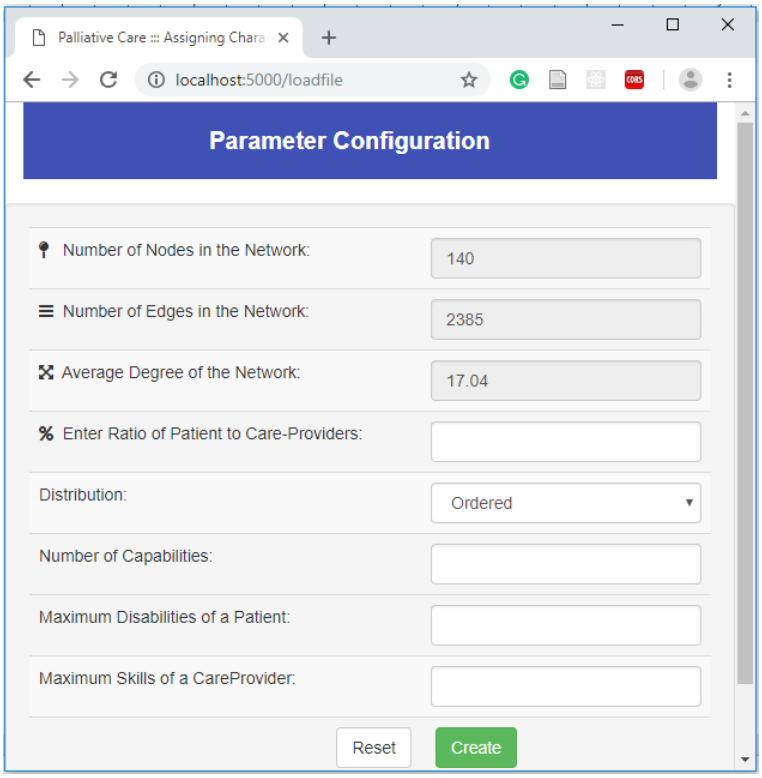


Figure 4.5: Assigning the required parameters to generate synthetic network.

viding any value; the system will automatically assign numerical values for each capability if the name is not provided.

iv Maximum number of missing capabilities for each Patient: The user can determine the maximum number of capabilities that a patient can miss.

v Maximum number of services for a care provider: The number of services that a care provider can offer to the network can be adjust in this field.

In case of loading the real-world data, as the entire profiles information are uploaded to the system, there is no need to assign the role and other features. Figure. 4.4 shows the interface for uploading profile details of the care team in a real network.

4.4.1 Data Visualization

The visualizing process of data will take the values from the parameter configuration page as we discussed in the previous section. Figure 4.6 shows the visualization of a given sample synthetic network. This tool enables the user to expand or shrink the network visualization, to view information of a specific node, and to search for any specific node and explore the raw network.

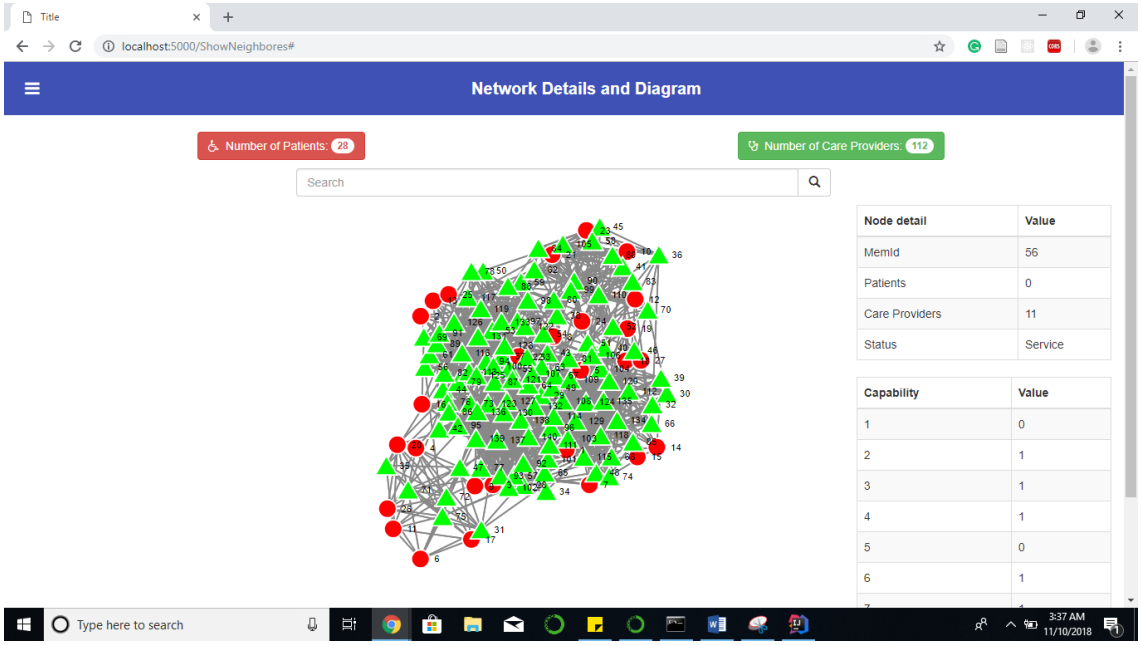


Figure 4.6: The visualization of palliative care - synthetic network.

As discussed before, visualization of a team of care is one of the main objectives of this framework. Figure. 4.7 shows the optimal team members of our sample network based on the given parameters. The sample synthetic network was generated with 140 nodes and assigned 20% as patients and 80% as care providers, number of capabilities/services set to 8, Maximum disability of a patient set to 30% and the maximum service that a care provider can offer was 60%. As a result of this visualization, anyone can easily observe which patient is getting service by which care teams, identify the fully occupied care providers and those ones who are not providing services, and whether every patient is getting services or not.

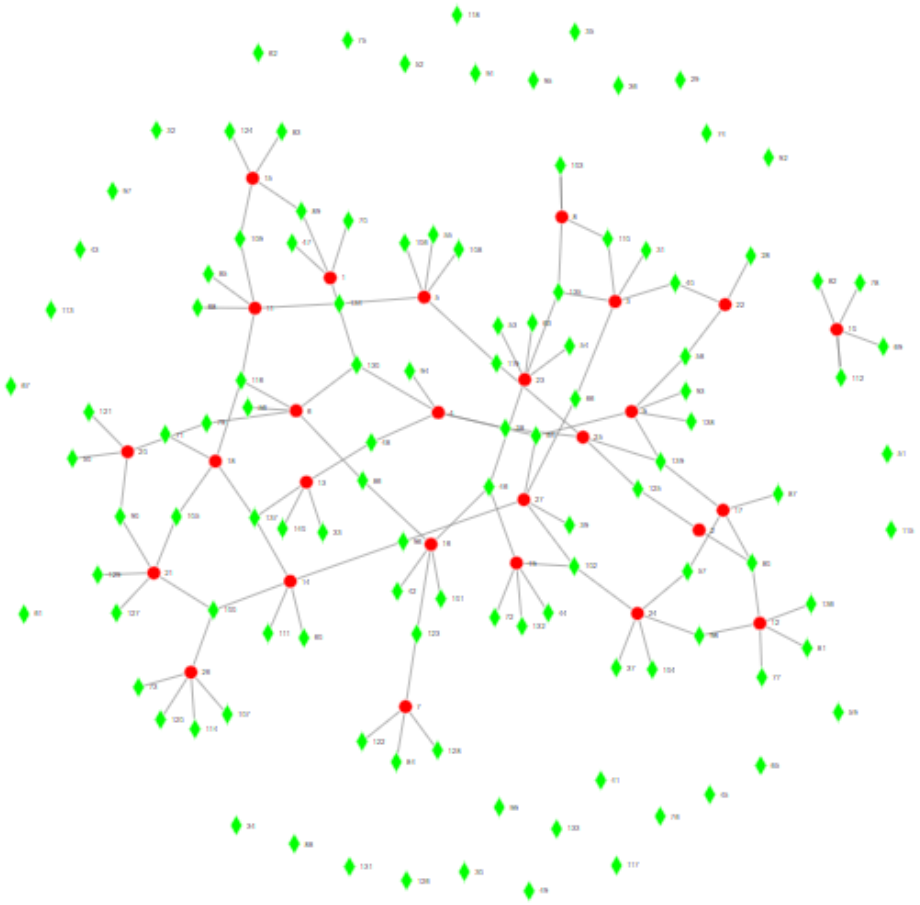


Figure 4.7: The care teams in our sample synthetic network with 140 nodes.

In addition, Figure.4.8 shows the communities and identified clusters in our tested network with 140 nodes. As discussed earlier, it is a very important tool that represents groups of nodes that have a high level of dependency to each other.

4.5 Conclusions

In this paper, we proposed a framework to generate and visualize structures and characteristics of palliative healthcare networks using both synthetic and real-world data. Our framework is capable of generating and customizing a wide range of care networks for the simulation purposes. It also is capable of identifying and visualizing clusters and efficient team of cares in a given network. This framework is useful for

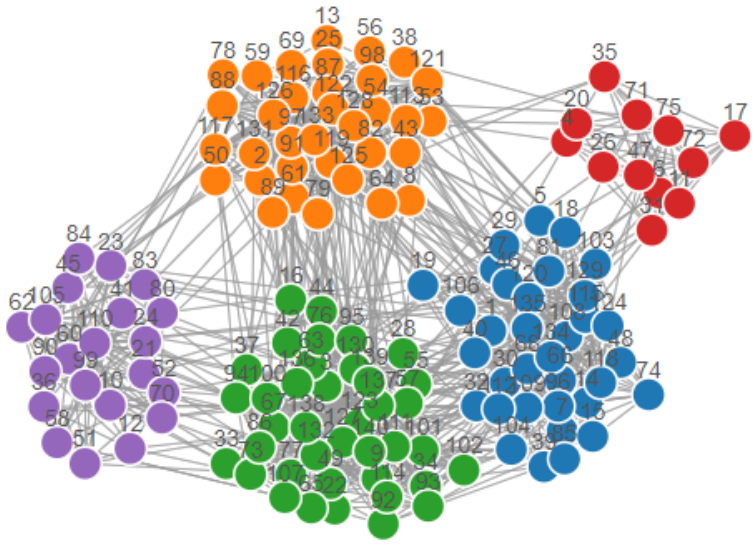


Figure 4.8: The cluster diagram of our sample synthetic network

the care providers and policy makers to explore the characteristics of the network and monitor its evolution during the time to identify hidden patterns among the system. In the future, we are going to add more features to this framework in order to represent spatio-temporal events in the network and update our predictive model.

Bibliography

[1] S. M. Mickan, "Evaluating the effectiveness of health care teams," *Australian Health Review*, vol. 29, no. 2, pp. 211–217, 2005.

[2] K. Selvarajah, P. M. Zadeh, Z. Kobti, M. Kargar, M. T. Ishraque, and K. Pfaff, "Team formation in community-based palliative care," in *2018 Innovations in Intelligent Systems and Applications (INISTA)*. IEEE, 2018, pp. 1–7.

[3] A. W. Crosby, *The measure of reality: Quantification in Western Europe, 1250-1600*. Cambridge University Press, 1997.

[4] T. Crnovrsanin, C. W. Muelder, R. Faris, D. Felmlee, and K.-L. Ma, "Visualization techniques for categorical analysis of social networks with multiple edge sets," *Social Networks*, vol. 37, pp. 56–64, 2014.

[5] N. T. Tam and I. Song, "Big data visualization," in *Information Science and Applications (ICISA) 2016*. Springer, 2016, pp. 399–408.

[6] C. Schikora and D. Isemann, "Influviz—a visualization tool for exploring and analyzing creative influence between artists and their works," in *Information Visualisation (IV), 2017 21st International Conference*. IEEE, 2017, pp. 336–343.

[7] K. Gemici and A. Vashevko, "Visualizing hierarchical social networks," *Socius*, vol. 4, p. 2378023118772982, 2018.

[8] D. Park, "Bom-vis: A visualization of network health and status."

[9] A. Avati, K. Jung, S. Harman, L. Downing, A. Ng, and N. H. Shah, "Improving palliative care with deep learning," *arXiv preprint arXiv:1711.06402*, 2017.

[10] N. Moradianzadeh, P. M. Zadeh, Z. Kobti, S. Hansen, and K. Pfaff, "Using social network analysis to model palliative care," *Journal of Network and Computer Applications*, vol. 120, pp. 30–41, 2018.

[11] A. Lancichinetti, S. Fortunato, and F. Radicchi, "Benchmark graphs for testing community detection algorithms," *Physical review E*, vol. 78, no. 4, p. 046110, 2008.

[12] P. M. Zadeh and Z. Kobti, "A Multi-population Cultural Algorithm for Community Detection in Social Networks," *Procedia Computer Science*, vol. 52, no. 9, pp. 342–349, 2015.

[13] R. G. Reynolds, "An introduction to cultural algorithms," in *Proceedings of the third annual conference on evolutionary programming*. World Scientific, 1994, pp. 131–139.

Chapter 5

Cultural Algorithms for Cluster Hires in Social Networks

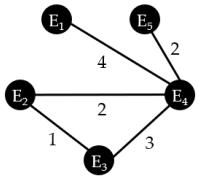
This research examines the team formation problems in a situation where the primary focus is to maximize the profit of projects under a given budget, called cluster hire problem. We can see the above scenario in the industry organizational setting and online freelancing jobs such as Upwork.com and Guru.com. This research attempts the knowledge-based cultural algorithms in a cluster hire problem and compares it with existing approaches.

5.1 Introduction

Team formation is a group of people with various skillsets coming together to form teams to complete certain tasks associated with certain constraints. Past working experiences bring compatibility among the members of teams. As a result, each member connects with others. Their connected community can be considered as social networks where each person represents a node, and their past relationship represents edges. In this study, we consider a collection of tasks that requires specific skillsets and produces a different profit upon completion. At the same time, each

Project	Skills	Profit
P_1	AI, ML, CN	180
P_2	DB,ML	240
P_3	AI,GR,WD	120

(a) Project Requirements



(b) Social Networks of experts with the communication cost between them.

Expert	Cost	Skills	Capacity	Productivity
E_1	25	AI, WD	2	4
E_2	20	ML,CN	3	3
E_3	30	GR, WD	2	4
E_4	15	AI, DB, ML	4	3
E_5	15	DB,GR	3	2

(c) Experts Details

Figure 5.1: Example of project requirements, a social Networks of experts with communication cost between experts and expert profile information.

person demands a certain amount of salary to invest their time and knowledge. A budget is required to spend on those expenses. Our primary objective is to identify a group of teams that maximizes the total profit of the tasks that can be completed under a given budget. This problem was first introduced by Golshan *et al.* [1], and called it a Cluster Hire problem.

We can observe the above similar settings in industries or online platforms such as Freelancer, Guru, and Upwork. In these settings, people with diverse skills can be hired to work on different types of projects. In the era of web 2.0, communication and transferring information through the internet becomes easier. Most people realize that teamwork has more advantages than working alone. So, the cluster hire problem is considered as a significant problem in the real world.

The primary concern of the standard cluster hire problem was to hire a profit-maximizing team of experts with the ability to complete multiple projects within a given budget [1]. Later some researches examined this problem with additional measuring parameters such as compatibility [2, 3], and productivity [4] in order to improve the accuracy of the solution. Existing researches applied greedy algorithm [1, 4, 3] and Linear Programming [2] to tackle this problem. Besides this, since people are connected through the network, capturing social network features are beneficial to solve the problem. In terms of influencing parameters, existing research failed to combine the social compatibility of team members, the productivity of teams, and the capacity of a person in a framework.

For example, let us assume that a social network has 5 people, and our objective is to complete 3 projects, as shown in Figure. 6.1. Required set of skills are artificial intelligence (AI), databases (DB), graphics(GR), web development (WD), computer networks (CN) and machine learning (ML). In our model, we want to find the teams of experts who satisfy the project requirements, maximizing profit as well as minimizing communication cost (minimum communication cost means that the compatibility among them is high). The project requirements can be satisfied with two teams: $A = \{E_1, E_2, E_5\}$ and $B = \{E_2, E_3, E_4\}$. Both A and B satisfy the capacity of individuals too. First of all, the connection between team members is important because it increases compatibility. The communication cost can be 6 and 12 for A and B, respectively. So, we select team A to perform these project lists to satisfy every constraint. This process is not simple with complex networks.

The existing approaches used greedy algorithms and linear programming with approximation algorithms. When the size of the network increases, these algorithms are not guaranteed to find the optimal solution within a considerable runtime. So, to avoid this issue, we employ knowledge-based cultural algorithms.

We make the following contributions in this paper:

1. We tackle the cluster hire problem by using a knowledge-based evolutionary optimization model for the industry based settings.

2. As an extension of the basic problem [1], we consider the compatibility between team members to collaborate with each member effectively.

3. We also consider the capacity of a person to a certain number, which means the number of projects can be handled by that person simultaneously.

4. We also include the productivity of the person in our model. The productivity measures the experience of a person in a specific skill.

Our previous work in [5] considered the case study for the academic collaborations and used social compatibility that is communication cost between team members to measure the effective teams. It measured the communication cost in two different methods: Diameter distance and Shortest path distance. It was a single objective optimization problem. On the other hand, this paper considers the case study in industry organizational settings. Since industries focus on profit, we maximize the profit by hiring efficient teams of experts under a given budget. We use three performance evaluation functions: communication cost, productivity, and workload to measure the performance of the teams, which decide the efficient teams of experts. This problem is a multi-objective optimization problem. The size of the solution or chromosome in this work varies based on the budget and the profit because finding a set of teams under the given budget might only be suitable to complete a few projects or whole projects from the project list. Therefore, implementing this process is highly complicated. This study has several benefits in the real world, such as in any business for profits and online freelancing jobs.

The rest of the paper is organized as follows. In the next section, we discussed the related work. We then present the problem statement in Section 3. In Section 4, we discuss the knowledge-based framework to find the best team of experts. In Section 5, a comprehensive set of experiments over synthetic data is presented. Finally, we conclude our model in Section 6.

5.2 Related Works

For the first time in social networks, Lappas *et al.* [6] introduced the team formation problem to minimize the communication cost. In the last two decades, there are several papers tackled this problem in various ways. Some of them modify the first proposed function to some advanced version such as enhanced Steiner algorithm [7] , shortest path method [8] and diameter distance, the longest shortest path among

Symbols used in this paper

$\mathcal{E}$	set of n experts $\{e_1, e_2, \ldots, e_n\}$
$\mathcal{S}$	set of m skills $\{s_1, s_2, \ldots, s_m\}$
$\mathcal{T}$	set of k project $\{t_1, t_2, \ldots, t_k\}$
$\mathcal{T}'$	sub set of project $\mathcal{T}$
$\mathcal{S}(e)$	set of skills possessed by expert e
$\mathcal{S}(t)$	set of skills required for project t
$\psi(e)$	cost of hiring expert e
$\Theta(e)$	load limit of expert e to offer his/her expertise
$\Delta(t)$	profit of completing project p
$\tau(e)$	productivity of a person e
$d(e_i, e_j)$	shortest distance between experts e_i and e_j
Γ	total budget for hiring experts

team members [9] while others incorporate few other parameters such as expertise level [9] and geographical proximity [10, 11]. At the same time, many of these models used the greedy algorithm and the approximate method.

Some algorithms applied the evolutionary algorithms in TFP. The authors [10] applied Genetic Algorithms and studied the geographical location of each member of the team while optimizing the approach. The authors [11] considered the TFP in the health care setting and applied Cultural algorithms to optimize multi-objectives.

Different from all the above works, [1] aimed to hire a profit-maximizing team of experts with the potential to complete multiple projects, within a given budget. It is a promising problem in real-world settings. Later [2] proposed a bit similar to Golson's [1] work by incorporating social compatibility, and they named their work as TEAMGROUPING. They applied Linear Programming with an approximation algorithm. The authors [3] proposed almost similar work to [2], but solved using greedy algorithms.

5.3 Problem Statement

In this paper, $\mathcal{E} = \{e_1, e_2, \ldots, e_n\}$ specifies a set of n experts, $\mathcal{S} = \{s_1, s_2, \ldots, s_m\}$ specifies a set of m skills and $\mathcal{T} = \{t_1, t_2, \ldots, t_k\}$ specifies a set of k projects. Each individual $e \in \mathcal{E}$ has a subset of skills, *i.e.*, $\mathcal{S}(e) \subseteq S$, the set of skills possesses by the individual e. Each project $t \in \mathcal{T}$ requires a subset of skills, *i.e.* $\mathcal{S}(t) \subseteq \mathcal{S}$, set of skill required to complete project t and $\Delta(t)$ be the profit from the project t. At the same time, each individual demands a pay $\psi(e)$ for each task t and associates with a number for the capacity $\Theta(e)$, the maximum number of project can be covered by the person e. In our setting, we incorporate the productivity of the person $\tau(e)$, to determine how best can person e work on project.

Definition 5.1. ***Group of Experts*** *For a set of n experts $\mathcal{E}$, a set of m skills $\mathcal{S}$, and a set of k projects $\mathcal{P}$, a group of experts $\mathcal{E}' \subseteq \mathcal{E}$ is expected to complete a subset of projects $\mathcal{T}' \subseteq \mathcal{T}$ with the following condition.*

Load: Each team member can have different load limits, *i.e.*, each person should not work on more than $\Theta(e)$ number of tasks at a time. This value can be decided based on the work history, how many projects each individual completed successfully in parallel.

5.3.1 Social Compatibility

Social compatibility plays a significant role to decide how comfortably each member of the teams collaborate to work together. We consider the relationship among these individuals from a social network $\mathcal{G}(\mathcal{E}, \mathcal{R})$, where $\mathcal{E}$ represents the set of individuals and $\mathcal{R}$ represents the relationship among them. Lappas *et al.* first introduced the concept of social compatibility [6] in TFP. The shortest distance between any two members can be calculated based on their past collaborations in the same project [8].

The shortest distance between any two members can be computed as:

$$W_{e_i,e_j} = 1 - \frac{|T_{e_i} \cap T_{e_j}|}{|T_{e_i} \cup T_{e_j}|} \tag{5.1}$$

where T_{e_i}(resp.T_{e_j}) is the set of collaboration by e_i (resp. e_j).

In our model, we adopt the concept of the shortest path distance to evaluate the social compatibility between team members [8]. It can be defined as below.

$$SumofDistances(\mathcal{E}') = \sum_{(s_i;s_j)\in P\times P} d(e_{s_i}, e_{s_j}) \tag{5.2}$$

Where $d(e_{s_i}, e_{s_j})$ indicates the possible shortest path length in the given social graph, which connects any two members of the teams such that one of them is assigned to cover the skill s_i and the other one is assigned to cover the skill s_j. Here, the smaller the shortest distance represents better social compatibility.

5.3.2 Productivity

The basic idea of productivity $\tau(e)$ of a person is to have the best expertise in the teams. It can be decided based on the number of task e performed in the past. So, the total sum of the productivity of the team needs to maximize to find better teams of experts.

$$Productivity(\mathcal{E}') = \sum_{(e_i\in\mathcal{E}')} \tau(e_i) \tag{5.3}$$

where $\tau(e_i)$ specifies the productivity value of any expert in the team.

5.3.3 Profit of the Projects

Generally, each project experts a certain level of profit. In our setting, the profit of a project represents as $\Delta(t)$. So, the profit of finishing a set of projects $\mathcal{T}$ can be defined as follow.

$$Profit(\mathcal{T}) = \sum_{(t_i \in \mathcal{T})} \Delta(t_i) \tag{5.4}$$

where $\Delta(t_i)$ specifies the profit of any project in $\mathcal{T}$.

We set the budget to Γ that helps to manage the expenses for the hiring process. Our primary goal is to assign a group of teams to perform a set of projects while the expenses should be under the given budget, and the profit should be maximum as possible. Overall, we aim to find a group of teams who have the minimum sum of distance and maximum productivity, and the group of teams should maximize the profit while they work on their capable capacity level. So, we can formulate our problem as a tri-objectives optimization problem. In this setting, we convert our problem into a single objective problem by introducing trade-off parameters α, β and γ such that $\alpha + \beta + \gamma = 1$.

Problem 5.1. *A set of n experts $\mathcal{E}$ with a set of m skills $\mathcal{S}$ need to complete a set of k projects $\mathcal{T}$. We assigned the trade off α, β and γ among the shortest path distance, profit and productivity. We aimed to choose a group of teams $\mathcal{E}' \subseteq \mathcal{E}$ and a set of projects $\mathcal{T}' \subseteq \mathcal{T}$ in which the following objective is maximized:*

$$CH(\mathcal{T}', \mathcal{E}') = (\alpha).(1 - SumofDistances(\mathcal{E}')) + (\beta).Profit(\mathcal{T}') + (\gamma).Productivity(\mathcal{E}') \tag{5.5}$$

Addition to this, our budget value must be satisfied the following condition: $\sum_{e \in \mathcal{E}} \psi(e) \leq \Gamma$, where $\psi(e)$ specifies the payment demanded from expert e and Γ specifies the total budget to complete $\mathcal{T}$ projects.

In equation 5.5, each parameter takes its own units. For example, profit will be at *dollars*, and the other two are just numbers. Therefore, we normalized them to have the same scale.

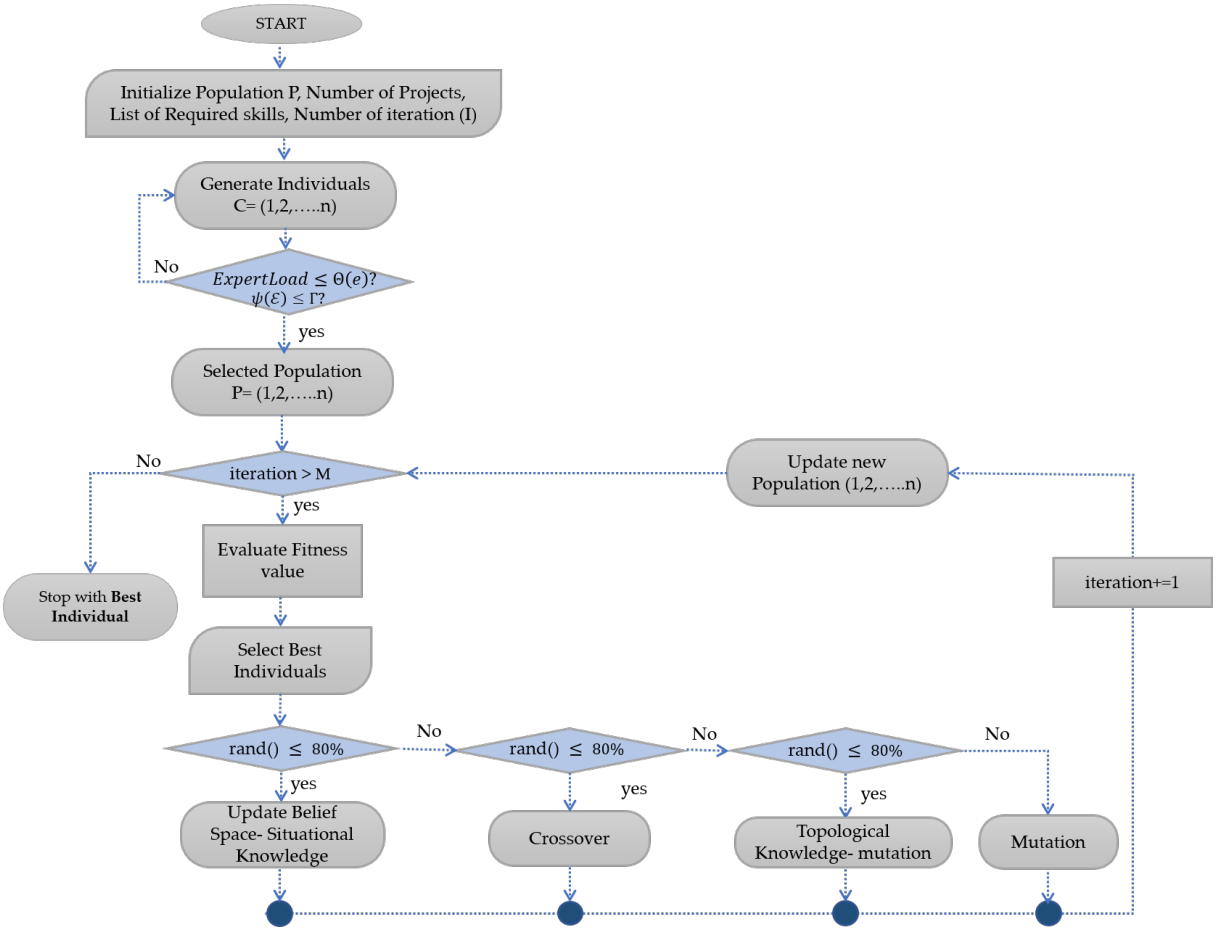


Figure 5.2: The flow chart to describe how knowledge-based algorithm handled to produce the best solution.

5.4 Evolutionary Algorithm

The cluster hire problem was proven to be an NP-Hard problem by Golshan *et al.* [1]. The existing similar problems used the methodologies which can optimally find the solution such as greedy and Linear Programming with approximation. We already discussed that a knowledge-based evolutionary algorithm tried to find an optimal solution even with complex networks. In this section, we discuss the evolutionary algorithm that we use in our proposed model.

We used knowledge-based cultural algorithms to test the model. The Cultural algorithm uses two phases: population space and believes space [12]. Similar to

other evolutionary algorithms, such as the Genetic algorithm, begins with the random initial population of chromosomes. The chromosome or individual means the possible solution to the problem. The fitness function can be used to evaluate the ability of individuals to survive for the next generation. The cultural algorithm uses some knowledge extractions such as situational, normative, and topological knowledge from selected individuals. The influence function involves knowledge extraction from the belief space, while acceptance function keeps the most suitable individual knowledge into belief space. This process helps to guide the search direction and continue until the termination condition. Finally, it returns the best solution to the problem.

5.4.1 Representation

The representation of the individuals is a significant task in evolutionary algorithms. In this setting, the size of the chromosome is dynamic because we have to find the number of projects which can be completed under the given budget. For example, if the number of projects $\mathcal{T}$ is 3, to find the subset of projects $\mathcal{T}'$, which can be completed within a given budget Γ. The chromosome sizes for this case can be $\{1, 2, 3\}$. Every chromosome size will keep all the combinations of the projects. Let's say, if we want to generate the chromosome size 2, in our case we have to create C_2^3 combinations of chromosomes, for example $\{t_1, t_2\}, \{t_1, t_3\}, \{t_3, t_2\}$ are the combinations of 3 tasks of 2 sub-tasks. In this case, we need to create 7 different size of chromosomes, as shown in Figure 5.3. In general, the number of combinations would be $2^{|\mathcal{T}|-1}$.

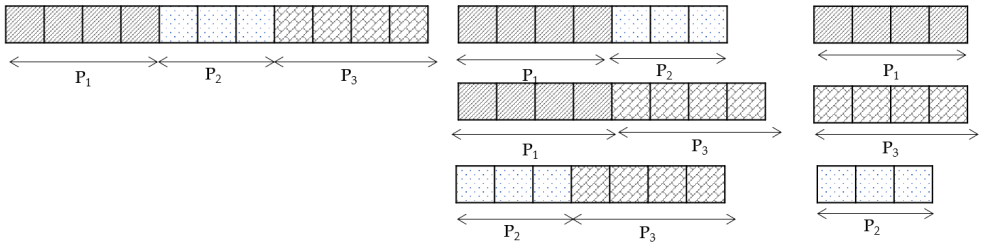


Figure 5.3: Example of chromosome structure when we test for the total number of projects 3.

5.4.2 Belief Space

The belief space is to utilize the extracted knowledge from selected individuals to enhance the performance of cultural algorithms. It updates the extracted knowledge to the next generations. In our setting, we extracted two knowledge: normative and topological knowledge. For the first case, we adopted the methodology used in [5] to extract the knowledge from the best individuals, while in the second case, we apply the topology for an unexplored list which we will explain in the following section.

5.4.3 Genetic Operators

As every evolutionary algorithm, we also apply recombination and mutation in our model. The recombination is carefully handled among the same subset of chromosomes. We applied a single-point crossover to generate offsprings of the subsets. For a certain percentage of chances, we perform recombination among the best individuals, while at other chances, it combines random two individuals. Similarly, we perform mutation with a random gene for some probability, while in other chances, we mutate the gene with the unexplored list that is the list of genes that are not used in the current populations.

5.4.4 Procedure

The flow chart in Figure.5.2 describes the step by step procedure of our method. We begin by generating a predefined number of random teams to create the initial population. While we create each chromosome, we make sure whether each expert exceeds their load limit or not and whether the budget Γ is greater than or equal to a total pay of the experts $\psi(E)$ of the chromosome. As explained in the above section, the remaining steps will perform genetic operations to create offspring.

The process stops when the best solution is found. The best solution can be a

set of required teams to solve all projects or a few projects. If it discovers teams for all projects, for instance, the size of the chromosome is 11 for all 3 projects, as we described in Figure 5.3. If the budget is enough to cover only a few projects, then the size of chromosome changes among $\{3, 4, 7, 8\}$ values, which represent the teams to complete either 1 or 2 projects. Our model decides the above result based on the given budget to maximize the profit.

5.5 Experiments

In this section, we test the performance of our proposed algorithms over a synthetic Social Network generated using the LFR benchmark [13]. We create a string array to keep a set of skills. We then assign a set of skills to each person randomly between 2 to 5. Based on the past collaborations in the network, we generate the shortest path distance among each member. We randomly assign the load limit of a person from 2 to 6, the productivity of a person 1 to 10, and the paid amount of a person from 30 to 50. We generate random projects for the user-defined number. Each project is assigned to a profit from 100 and 6000. We run each experiment for 5 times and evaluate the average value. For the objective function, we assign the following trade-off values: $\alpha = 0.3, \beta = 0.4$ and $\gamma = 0.3$.

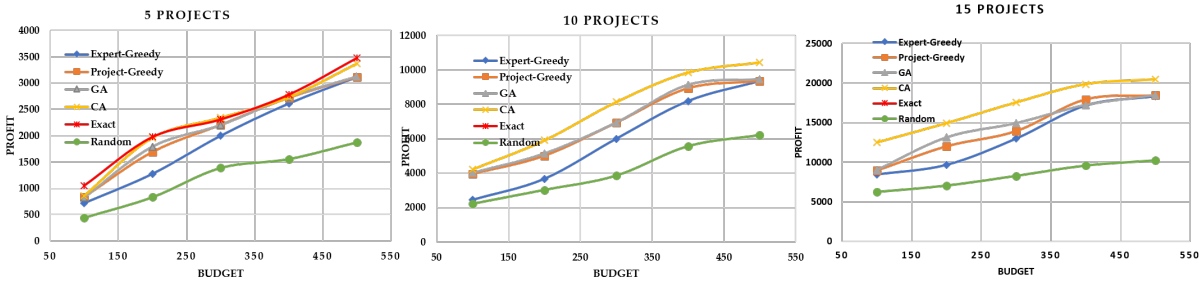


Figure 5.4: Comparison for Total profit vs. budget of CA , GA, Project Greedy, Expert Greedy,Exact and Random Algorithm

We compare our method with project greedy, expert greedy algorithms as pro-

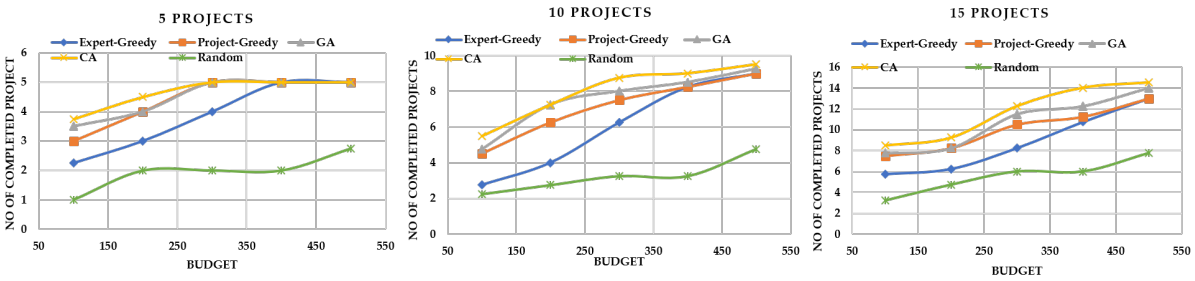


Figure 5.5: Comparison for Total profit vs. budget of CA, GA, Project Greedy, Expert Greedy, Exact and Random Algorithm

posed in [4], exhaustive algorithm, and random search method. We plot the graph for the budget vs. profit (Figure. 5.5)and budget vs. the number of completed projects (Figure. 5.4), in which except random method, every other algorithm perform well for the small number of projects. However, when the number of projects increases, the CA performs better than other algorithms. Moreover, from the graph, we can observe that most of the time, GA and Project greedy produce almost similar results. Although expert greedy did not perform well for the less budget, it equally performs to project greedy for the high budget rate.

In terms of runtime, the CA takes little more time than GA but better than project greedy algorithm. Because CA extracts knowledge and updates to the next generation, the processing time takes a little longer than GA. Moreover, the run time of expert greedy was lower than project greedy algorithm.

5.6 Conclusions

In this paper, we tackled the cluster hire problem, which was introduced by [1] using the knowledge-based cultural algorithm. At the same time, we set significant measuring parameters: social compatibility, load limit, productivity, and profit of the projects to find the group of teams. We formalize this problem as a tri-objectives problem. Then it is converted to a single-objective problem by introducing trade-off

parameters. We test our model on synthetic social networks. Then we compare it with other recently proposed algorithms such as project greedy and expert greedy algorithms [4]. We also compared them with the random and exact algorithm. Our approach outperformed against other compared methods. In the future, we would like to test our model on real-world datasets such as Upwork and Freelancer.

Bibliography

[1] B. Golshan, T. Lappas, and E. Terzi, "Profit-maximizing cluster hires," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining.* ACM, 2014, pp. 1196–1205.

[2] S. Tang, "Profit-driven team grouping in social networks," in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[3] S. Liu and C. K. Poon, "A simple greedy algorithm for the profit-aware social team formation problem," in *International Conference on Combinatorial Optimization and Applications.* Springer, 2017, pp. 379–393.

[4] P. Patel, K. Selvarajah, Z. Kobti, and M. Kargar, "Productive and profitable cluster hire," in *The Thirty-Second International Flairs Conference*, 2019.

[5] K. Selvarajaha, P. M. Zadeha, M. Kargarb, and Z. Kobtia, "Identifying a team of experts in social networks using a cultural algorithm," *Procedia Computer Science*, vol. 151, pp. 477–484, 2019.

[6] T. Lappas, K. Liu, and E. Terzi, "Finding a team of experts in social networks," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining.* ACM, 2009, pp. 467–476.

[7] A. Anagnostopoulos, L. Becchetti, C. Castillo, A. Gionis, and S. Leonardi.,

"Power in unity: Forming teams in large-scale community systems," in *Proc. of CIKM'10*, 2010.

[8] M. Kargar and A. An, "Discovering top-k teams of experts with/without a leader in social networks," in *Proceedings of the 20th ACM international conference on Information and knowledge management*. ACM, 2011, pp. 985–994.

[9] A. Anagnostopoulos, L. Becchetti, C. Castillo, A. Gionis, and S. Leonardi, "Online team formation in social networks," in *Proc. of the WWW'12*, 2012.

[10] Y. Han, Y. Wan, L. Chen, G. Xu, and J. Wu, "Exploiting geographical location for team formation in social coding sites," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2017, pp. 499–510.

[11] K. Selvarajah, P. M. Zadeh, Z. Kobti, M. Kargar, M. T. Ishraque, and K. Pfaff, "Team formation in community-based palliative care," in *2018 Innovations in Intelligent Systems and Applications (INISTA)*. IEEE, 2018, pp. 1–7.

[12] R. G. Reynolds, "An introduction to cultural algorithms," in *Proceedings of the third annual conference on evolutionary programming*. World Scientific, 1994, pp. 131–139.

[13] A. Lancichinetti, S. Fortunato, and F. Radicchi, "Benchmark graphs for testing community detection algorithms," *Physical review E*, vol. 78, no. 4, p. 046110, 2008.

Chapter 6

A Unified Framework for Effective Team Formation in Social Networks

The significant objective of this chapter is to propose a unified framework for Team Formation Problem (TFP) to solve any application in dynamic social networks. We examine several measuring parameters to decide efficient teams and formulate it as a multi-objective optimization problem.

6.1 Introduction

In numerous circumstances, the use of team-based opportunities has become more common. Such teams are often confined within a group or society, especially in the manufacturing, law, academia, healthcare, in freelancer jobs such as Upwork and Guru, and other professional organizations. In most of these situations, people are connected as networks based on past collaboration or colleagues. These connected networks can be considered as social networks that contain professionals as nodes who have a set of skills, expertise in particular areas, and the relationships between

pairs of individuals being the edges.

Team formation is a group of individuals coming together to form a team to work on a task with expectations that the task should be completed successfully. TFP in social networks is to collect the group of individuals who match the requirements of given tasks. It is a challenging process because it needs to ensure that the team assembled to carry out a task effectively. Many researches have been conducted to solve the TFP optimally. However, this research requires new approaches to improve accuracy. To discover the productive or successful teams to work on projects, satisfying the requirements of the projects are not sufficient. It requires several other details related to the individuals in the teams, such as how they are connected as a social network and what is the relationship between them.

Most of the existing models have considered communication cost among teams because the minimum communication cost delivers their tasks more efficiently. Lappas *et al.* [1] proved that the communication cost has a higher impact when forming the teams in social networks. Later many researches [2, 3, 4] modified their approach in various ways such as enhanced Steiner algorithm and shortest path distance. Some other researches have been conducted to analyze the different parameters which influence the formation of teams such as expertise score [4, 5], geographical proximity [6], and density [7].

Existing researches ignored to analysis the expertise as a dynamic score. Let us consider an example; if the expert profile state that he or she has a list of skills, it does not mean that he or she is an expert in every skill. Besides, in reality, people never keep the same expertise level all the time, if they do not work on it. So, based on how often anyone works on any skill, and when did they contribute their effort on a specific skill, we can decide the current level of expertise as a dynamic score. Moreover, although existing researches examined the communication cost of the team using different approaches, they failed to discuss the dynamic nature of it. Bringing

the temporal features of the expertise score and communication score is significant in TFP. Therefore, we introduce two formulas to incorporate the dynamic behavior of expertise score and communication cost in our framework.

A team without trust is just a group of individuals working together, often making unsatisfactory progress. This study considers three factors to measure trust quantitatively: Explicit trust score, Profile similarity score, and Emotional intelligence index. We believe that a direct trust score is not only sufficient information to evaluate trust between any two members in a team because it might be biased. Scores which tell how much a person has similar interest with others and whether a person emotionally fit to work as a team or not are also significant information to calculate the trust score. Although these scores are not quantifiable to construct a computational model, we bring some ideas in this study based on the researches in management.

Our primary objective is to model a unified framework for team formation problem in social networks to analysis various contexts which associate with each member of a team: how frequently have team members work in the past, are they emotionally fit to work as a team, how much each team member have faith in other members and are they living close enough. To address these questions, we design a unified team formation model to study the formation of teams and their effect on production.

The remaining of this paper is organized as follows. We discuss related works in Section II. Section III is to define the problem definition for the TFP problem in a dynamic environment. Then we discuss the Multi-Objective Cultural Algorithm for TFP in Section IV. In Section V, we conduct extensive experiments on synthetic datasets. Finally, we conclude this paper in Section VI.

6.2 Related Works

Lappas *et al.* [1] first addressed the Team Formation Problem (TFP) with social networks. They formulated TFP using communication costs and proved that it was an NP-hard problem. In the last decade, TFP got a great deal of attention from many researchers [2, 8, 9, 10]. In addition to communication costs, some studies examined other aspects of contexts to form successful teams, including workload, expertise level, personnel cost, density, geographical proximity, and trust score.

6.2.1 Communication Cost

The concept of communication cost (CommCost) is used to measure the effectiveness of collaboration within a team. Lappas *et al.* [1] first suggested that it can be calculated using social network analysis, based on the interactions of the individuals. The success of a project relies on how well the experts in teams communicate and collaborate with others. The CommCost measures the closeness of the individuals in a social network $\mathcal{G}$. If two individuals V_i and V_i are adjacents, the CommCost is the weight of the edge (V_i, V_j); otherwise, the CommCost is the shortest path between V_i and V_i. Numerous studies focused on the concept of CommCost. These studies discussed various definition including minimum spanning tree [1, 11, 4, 12, 13, 14, 15], the weight cost of the minimum spanning tree for subgraph formed by a team, diameter distance [1, 15, 10], the longest shortest path between any individuals in a team and sum of shortest distance [2, 16, 15, 17, 18, 19, 10], the sum of all shortest paths between any two individuals in a team.

6.2.2 Work Load

In the multi-project situation, adjusting the amount of work according to the capacity of the individual is an essential factor to complete the work successfully. Some

Authors	CommCost	Work Load	Expertise Level	Personnel Cost	Density	Geo-Proximity	Trust Score
[1, 2, 11], [16, 13, 14], [15, 18, 19], [10]	✓						
[4, 12, 20]	✓	✓					
[3, 17, 21]	✓			✓			
[22]	✓	✓				✓	
[23, 24]	✓					✓	
[25, 5, 26], [27, 28]	✓		✓				
[29]			✓				
[30]			✓				✓
[31]							✓
[20, 32, 33], [7]					✓		

Table 6.1: Team Formation Problem based on various Parameters

researches [4, 12, 20, 22] focused the load balance on formulating the TFP in Social Networks.

6.2.3 Expertise Level

In the real world, knowledge structures of skills are quite diverse. Many people have different knowledge and can be experts at a different level. As specialization in every field increases, an individual who is an expert, has to maintain the level of expertise in specialized skill in a certain discipline to perform a given task skillfully. In a given extensive social network G, many individuals might have the ability to perform a specific task. Some people perform at a higher level than the others. So, there should be a method to measure the expertise score in any particular skill to find a suitable candidate to complete the task.

The skill mastery level of the whole team decides the successful outcome of the project. Many studies focused on the expertise level and considered it as a binary value: if a person has a skill, it is 1; otherwise, it is 0. These studies combined

the expertise level with the communication costs [25, 5, 26, 27, 28]. Some studies examined the expertise level with another context, such as trust [30].

6.2.4 Personnel Cost

In many real situations such as Guru.com, Freelancer.com, and Upwork, people work for pay, which is the personnel cost of experts. Finding teams with affordable cost is a benefit for the projects desirable to find a team of experts with a reasonable personnel cost [34]. The authors in [3, 17, 21] modeled the TFP with the combinations of CommCost and Personnel Cost.

6.2.5 Geological Proximity

Collaboration and the exchange of knowledge are easier by geographical proximity. Although many studies in social science theoretically discussed the benefits of geographical proximity in collaboration of individual, limited literature focused on computer science researches [23, 24, 22]. In the era of web 2.0, some can argue that geological proximity does not matter to work as a team. However, the chances of collaborations are high when experts live close.

6.2.6 Trust Score

Team trust is increasingly being recognized as essential for team performance. It has been generally agreed that a team with high collective trust is more successful than a low trust team [35]. Working in a team without trust, it is just a group of individuals. Research shows that in high-trust environments, people do their best work and are motivated to produce successful outcomes. If the team members make mistakes, everyone else in the team support and share information to produce the task effectively and efficiently.

The trust score gives the level of trust that members of the team have with each other. Because it becomes an important element of the success of social networks, many research has been examined in various perspectives [36]. Quantifying the trust value is a challenging process. The authors of [37, 30, 31] formulated trust score for the TFP. Awal et.al [30] used a ratings-based approach adopted from [38], which is based on the interactions that they had earlier.

6.2.7 Density

In a given graph G, the density method finds the densest subgraph, which satisfies the skill requirements of a project. The density of a team graph was examined in [32, 33]. Recently, an improved method was proposed in [7].

Table 6.1 summarize the existing related works in TFP based on various parameters to formulate successful teams. Many of these optimized the TFP as bi-objectives or tri-objectives by considering two or three parameters.

Many research papers ignored an important parameter in some way. For example, if the author proposed a model using both communication cost and workload, they failed to analysis expertise level or trust between team members. Moreover, the communication cost and expertise level do not remain the same with the time. For instance, if a programmer who is an expert in Java, got promoted and mentoring team members, or working in a different programming language such as Python, his expert level in Java will not remain the same all the time. In this regards, we incorporate the temporal behavior when evaluating the expertise score.

Similarly, with the communication cost, if two experts had frequent collaborations in the past years, and in recent years they don't collaborate, the chances of collaborating in the future are less. No one consider this in to account to model TFP with dynamic communication score. We focus these issues and propose a new unified framework for TFP by incorporating dynamic communication cost and dynamic

expertise score.

6.3 Proposed Model

Our model aims to find experts with the highest level of expertise who can perform the tasks, which require a certain expertise level in a cost-effective manner. To address the inherent cost of team members in various aspects, we have to demand a new model for the Team Formation Problem(TFP). This section will discuss the proposed comprehensive model for the TFP and outline key definitions to formulate our model.

6.3.1 Preliminaries

Given a social network $\mathcal{G}(\mathcal{V}, \mathcal{E})$ be an undirected weighted graph, where $\mathcal{V}$ represents a set of individuals or experts and $\mathcal{E}$ represents the relationship between them. The weights of the edges $\mathcal{E}$ in graph $\mathcal{G}$ can be interpreted as a indicator to measure how efficiently the individuals work as a team. We assume that there is a set of n skills $\mathcal{S} = \{S_1, S_2, \ldots, S_n\}$, a set of m individuals $\mathcal{V} = \{V_1, V_2, \ldots, V_m\}$. Each individual V_j associated with a set of skills $\mathcal{S}_i$ is a subset of skills, $\mathcal{S}_i(V_j) \subseteq \mathcal{S}$. Each skill in $\mathcal{S}_i$ is associated with a score based on the expertise level of the expert in a specific skill.

We assume that a set of k tasks is $\mathcal{T} = \{t_1, t_2, \ldots, t_k\}$. A task t_i is simply a set of skills $\mathcal{S}_j$ required to perform a project, i.e $\mathcal{S}_j(t_i) \subseteq \mathcal{S}$. Similarly as expertise score of expert in every skill, required skills set for a project is also associated with a score based on the required expertise level of a skill for that project.

Definition 6.1. ***(Team of Experts)*** *For a given set of experts $\mathcal{V}$ and a given task T that requires a set of skills $\{S_{i_1}, S_{i_2}, \ldots, S_{i_r}\} \subseteq \mathcal{S}$, a* team of experts *for t_i is a set of r skill-expert pairs: $t_i = \{\langle S_{i_1}, V_{j_1}^{S_{i_1}}\rangle, \langle S_{i_2}, V_{j_2}^{S_{i_2}}\rangle, \ldots, \langle S_{i_r}, V_{j_r}^{S_{i_r}}\rangle\}$.*

where $V_{j_p}^{S_{i_q}}$ is any expert $V_{j_p} \in \mathcal{V}$ who posses skill $S_{i_q} \in \mathcal{S}$, $\{V_{j_1}, V_{j_2} \ldots V_{j_r}\} \subseteq \mathcal{V}$ and $S_{i_q} \in \mathcal{S}$.

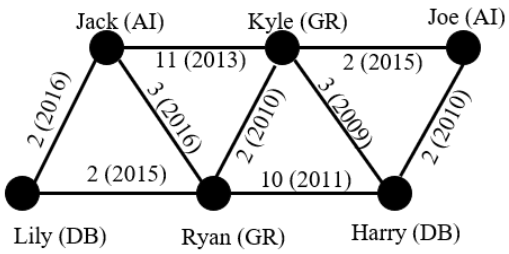


Figure 6.1: A social Networks of experts with expertise skill, the communication cost between them and the time when they did last project together.

Definition 6.2. ***(Tasks)*** *For a given set of tasks $\mathcal{T}$, if a task is represented as $t_i = \{\langle \mathcal{S}(t_i), V(t_i)\rangle\}$, then the set of task $\mathcal{T} = \{\langle \mathcal{S}(t_1), V(t_1)\rangle, \langle \mathcal{S}(t_2), V(t_2)\rangle \ldots \langle \mathcal{S}(t_k), V(t_k)\rangle\}$, such that $\sum |V(t_i)| = |\mathcal{V}'|$.*

6.3.2 Communication Cost

Kargar *et al.* [2] proposed Sum of Distances to measure the communication cost of a team. The distance in their work was the shortest path distance between the experts for each pair of skills. The expectation was that finding the team with minimum communication cost would finish the project on time.

- **Sum of Distance:** *Given team of experts $\{\langle S_{i_1}, V_{j_1}^{S_{i_1}}\rangle, \langle S_{i_2}, V_{j_2}^{S_{i_2}}\rangle, \ldots, \langle S_{i_r}, V_{j_r}^{S_{i_r}}\rangle\}$, the sum of distance between V_{j_p} and V_{j_q} of every pairs in the team can be defined as.*

$$SumDist = \sum_{p=j_1}^{j_r} \sum_{q=j_2}^{j_r} dist(V_p, V_q) \tag{6.1}$$

  where $dist(V_p, V_q)$ is the shortest distance between V_p and V_q in G. *i.e.*, the sum of weights on the shortest path between V_p and V_q.

Note that for the above cost function, the distance between two experts is defined over the whole graph G as a static parameter. In reality, the communication cost shows the dynamic nature and changes over time. Therefore, we enhance the above cost function 6.1 in order to incorporate the dynamic form of communication cost.

Time difference from current to last collaboration on a project is incorporated with the shortest path distance to adopt the temporal feature.

Definition 6.3. ***Dynamic Communication Cost:*** *Given team of experts* $\{\langle S_{i_1}, V_{j_1}^{S_{i_1}} \rangle, \langle S_{i_2}, V_{j_2}^{S_{i_2}} \rangle, \ldots, \langle S_{i_r}, V_{j_r}^{S_{i_r}} \rangle\}$*, the sum of distance between* V_{j_p} *and* V_{j_q} *of every pairs in the team can be defined as.*

$$DCC = \sum_{p=j_1}^{j_r} \sum_{q=j_2}^{j_r} (dist(V_p, V_q) + \alpha(t' - t)) \tag{6.2}$$

where $dist(V_p, V_q)$ is the shortest distance between V_p and V_q in G. t' is the current time and last time he or she work on a task together, and α is an attenuation factor.

For example, to motive the concept of dynamic communication cost, we illustrate a social network in Figure. 6.1. Assume that we need to perform a project which needs expertise in three areas: artificial intelligence (AI), databases (DB), and graphics (GR). We can form two teams $A = \{Jack, Lily, Ryan\}$ and $B = \{Kyle, Joe, harry\}$. The communication cost of both A and B is the same 7 if we do not consider the time of their last collaboration. On the other hand, if we consider the last collaboration time, we can evaluate the communication cost using equation 6.2 as below;

$$DCC_A = (2 + 0.1(2019 - 2016)) + (3 + 0.1(2019 - 2016)) + (2 + 0.1(2019 - 2015))$$

where we assume that the current year as 2019 and $\alpha = 0.1$. The dynamic communication cost of teams A and B would be 8.0 and 9.3, respectively. Smaller communication cost represents better team of experts who had frequent experience. Therefore, team A has higher chances to collaborate in the future.

6.3.3 Team Skill Mastery Level

The limited number of researches [30, 5] examine the expertise level in TFP. However, throughout the literature, this scale has been interpreted as a static score. Expertness

of an individual never remains the same if he or she is not work on a specific field. Considering the temporal behavior of the expertise score has significant influence on completing a task successfully. Past experiences are used to obtain the expertise score of an individual. Time difference from current to last work on specific skill is incorporated with the expertise score to adopt the temporal feature.

Experts	**Skills of projects**	**Worked Year**
Jack	AI	2009, 2011
Kyle	GR	2009,2010
Ryan	GR	2015,2017
Harry	DB	2009, 2010, 2011
Joe	AI	2014, 2017
Lily	DB	2016, 2015
Ryan	AI	2013

Table 6.2: The history of skills by each expert in a given SN G.

The expertise score of an expert $v_i \in \mathcal{V}$ can be defined as the ratio between the total number of projects or published papers $|ES_{V_i}^{S_j}|$ of the expert V_i in a skill $S_j \in \mathcal{S}$ and the total number of projects or published papers of the experts in a given network G in the skill S_j:

$$\mathcal{ES}_{V_i}^{S_j} = \frac{|ES_{V_i}^{S_j}|}{\sum_{p=1}^{r} |ES_{V_p}^{S_j}|} - \alpha(t' - t) \tag{6.3}$$

where r is the number of experts who interact with projects or papers with the skill S_j, t' is the current time, and last time he/she works on a task with skill S_j. α is an attenuation factor.

For example, let us consider the network of 7 people and the history of skills by each expert in a given SN G, as shown in table 6.2. The expertise score of *Jack* and *Joe* for *AI* is the same value 0.4 if we do not consider the temporal nature. But their score is different with temporal nature, 0.24 and 0.36 respectively if the current year is 2019 and $\alpha = 0.02$.

Definition 6.4. ***(Team Skill Mastery Level)*** *Given network G, if the expertise*

level of an expert $V_i \in \mathcal{V}$ in skill S_j is $\mathcal{ES}_{V_i}^{S_j}$, collective expertise level of a team is the sum of the expert level (ExpLevel) of the team of experts $\{\langle S_{i_1}, V_{j_1}^{S_{i_1}}\rangle, \langle S_{i_2}, V_{j_2}^{S_{i_2}}\rangle, \ldots, \langle S_{i_r}, V_{j_r}^{S_{i_r}}\rangle\}$, can be defined as,

$$EL = \sum_{p=j_1, q=i_1}^{i_r, j_r} \mathcal{ES}_{V_p}^{S_q} \tag{6.4}$$

6.3.4 Geographical Proximity

Geographical proximity plays a more subtle and indirect role in influencing collaboration and knowledge exchange [39]. Experts who located closely with others have high chances to collaborates due to many reasons such as cultural background, language, and time zone [24]. Therefore, considering geographical information will be useful for an effective team formation.

Definition 6.5. ***(Geographical Proximity)*** *The Geographical Proximity (GeoCost) of the team of experts $\{\langle S_{i_1}, V_{j_1}^{S_{i_1}}\rangle, \langle S_{i_2}, V_{j_2}^{S_{i_2}}\rangle, \ldots, \langle S_{i_r}, V_{j_r}^{S_{i_r}}\rangle\}$, can be defined as the sum of geographical distance between V_{j_p} and V_{j_q} of every pairs in the team.*

$$GC = \sum_{p=j_1}^{j_r} \sum_{q=j_2}^{j_r} geodist(V_p, V_q) \tag{6.5}$$

where $geodist(V_p, V_q)$ is the geographical proximity between V_p and V_q in G.

6.3.5 Trust Score

The trust mechanisms concern trust of an individual has in another, and have been proposed over the years. The trust score can be an explicit value that an individual directly gives a score to another based on their experience, such as in Epinion.com [40, 41]. The explicit value might be biased. In addition to this, if two individuals never interact in the past, there should be a mechanism to infer the trust between them. Moreover, if the individual A trust individual B, this does not imply that the

individual B should trust A. In other words, the property of the trust is asymmetry [42]. At the same time, trust does not hold the transitivity property. i.e., if a person A trust person B and B trust person C, we can't imply that A trust C.

To compute trust between any two individuals in a given network G, we decided to consider three information: Explicit trust score, Profile similarity score, and Emotional intelligence index.

Explicit Trust

The concept of explicit trust score is simple, and it is just the value given by an individual to another. It can be an integer between -1 and 1. Base on the people's experience, they can rate the trust value. If a person A trust person B, trust can be 1, if he/she does not trust, it can be -1, and if they have never interacted in the past, it can be 0.

$$ExpT_{v_i,v_j} = tr \tag{6.6}$$

where $tr \in \{-1, 0, 1\}$.

Profile Similarity

Many social psychological researches [43, 44] addressed that people with a similar taste like to communicate and work together. Further, Ziegler *et al.* [45] investigated the relationship between trust and profile similarity, and introduce a framework to quantify the trust using profile similarity when other trust evidence is absent.

In our research, because we primarily focus on expert networks, we consider the expert's skill descriptions to calculate the profile similarity. If anybody knew to what degree every expert is a specialist in each field, they could potentially utilize this learning to discover researchers with trust and recommend future collaborations [46]. The profile similarity between two experts can be calculated by taking into account

the relationship between any pair of skills, where a pair is formed by elements from the corresponding expertise expert list. For each such pair, a suitable value $\mathcal{V} \in [0, 1]$, reflecting the strength of the relation between the two concepts, can preliminary be associated with it. For example, $\mathcal{V} = 1$, when the corresponding concepts are identical and respectively, $\mathcal{V} = 0$ if they are not related. Thus the expertise similarity $\mathcal{SS}_{i,j}$ between two expert profiles, a_i and a_j, can be defined by equation 6.7.

Definition 6.6. ***(Score of Skill Similarity)*** *Given a network $\mathcal{G}\langle\mathcal{V}, \mathcal{E}\rangle$ and a set of experts $V = \langle v_1, v_2, \ldots, v_m\rangle$ has a set set of skills $\mathcal{S} = \langle s_1, s_2, \ldots, s_n\rangle$. The skill set of any two experts v_i and v_j can be $\mathcal{S}_{v_i} = \langle v_i^{s_{p_1}}, v_i^{s_{p_2}}, \ldots, v_i^{s_{p_r}}\rangle$ and $\mathcal{S}_{v_j} = \langle v_j^{s_{q_1}}, v_j^{s_{q_2}}, \ldots, v_j^{s_{q_l}}, \rangle$ respectively. If $v_i^{s_{p_x}}$ and $a_j^{s_{q_y}}$ are same skills $\mathcal{V} = 1$, otherwise $\mathcal{V} = 0$.*

$$ProSim_{v_i,v_j} = \sum_{k=1}^{n} \frac{\mathcal{V}_k}{n} \tag{6.7}$$

where n is the total number of expert's skill list of v_i and v_j, $n = |S_{v_i} \cup S_{v_j}|$.

Let's consider an example that an expert A has a list of skills $\{ML, DB, CN, WM\}$ and another expert B has a list of skills $\{ML, DB, SE\}$. The profile similarity will be 0.4, because the total number of skills is $|S_A \cup S_B| = 5$ and number of common skills is $|S_A \cap S_B| = 2$.

Emotional Intelligence Index

Emotional intelligence (EI) of a person is playing a key role in team performance, display the mental experiences, or deploying actual human behaviors. Management researcher[47] has paid great attention to EI. Emotional intelligence fosters trust, which can be built through EI [48]. In this study, we incorporate the EI index with the trust score.

Emotional intelligence can be defined in several ways, including the ability to understand emotions in oneself and others to make decisions, solve problems, and

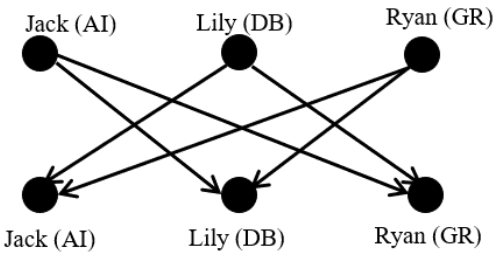


Figure 6.2: The representation for the method to calculate collective trust score of the team A.

communicate with others. It requires a strenuous effort to quantify as a score. To measure EI, many studies have used standard models or metrics such as Myers–Briggs Type Indicator (MBTI) and Five-Factor Model (FFM) and discovered significant associations between personality factors and the way of thinking [49]. These model give a quantified score for emotional intelligence.

Computing Final Trust Score

This section aims to return a final trust score for two connected individuals on social networks. As we discussed in the above sections, the final score is the combination of explicit trust score, profile similarity and emotional index. The following equation 6.8 provides how fully the expert v_i trust on v_j.

$$Tr_{V_i,V_j} = \alpha_1 ExpT_{V_i,V_j} + \alpha_2 ProSim_{V_i,V_j} + \alpha_3 EI_{V_j} \tag{6.8}$$

where EI_{v_j} will be the emotional index of the expert v_j. α_1, α_2 and α_3 are three balancing factors such that $\alpha_1 + \alpha_2 + \alpha_3 = 1$. Note that, the trust expert v_j has on v_i never be same as per the property of asymmetry.

Definition 6.7. ***(Collective Trust Score)*** *Given network G, if the trust of an expert $V_i \in \mathcal{V}$ has on $V_j \in \mathcal{V}$ is Tr_{v_i,v_j}, and V_j has on V_i is Tr_{V_j,V_i}. Note that $Tr_{v_i,v_j} \neq Tr_{V_j,V_i}$. The collective trust score of a team is the sum of the trust score (CT) of the team of experts $\{\langle S_{i_1}, V_{j_1}^{S_{i_1}}\rangle, \langle S_{i_2}, V_{j_2}^{S_{i_2}}\rangle, \ldots, \langle S_{i_r}, V_{j_r}^{S_{i_r}}\rangle\}$, can be defined*

as,

$$CT = \sum_{p=j_1}^{j_r} \sum_{q=j_2}^{j_r} Tr_{V_p,V_q} + \sum_{p=j_1}^{j_r} \sum_{q=j_2}^{j_r} Tr_{V_q,V_p} \tag{6.9}$$

As shown in Fig 6.2, collective trust of the team consider the trust between every two pair of individuals separately. For example, the trust between *Jack* and *Lily* consider $Tr_{jack,lily}$ and $Tr_{lily,jack}$, where $Tr_{jack,lily} \neq Tr_{lily,jack}$.

6.4 Multi Objective Cultural Algorithm for TFP

The proposed TFP framework is a multi-objective optimization problem. Here, every objective is equally important to the problem. Applying the trade-off method is favor some objectives to other objectives. Therefore, to solve the multi-objective TFP, we use the Multi-Objective Cultural Algorithm (MOCA) Framework. It is important because solutions to TFP are regarded from a variety of perspectives and cannot be expressed using only one objective.

The MOCA is the extended version of Cultural algorithms (CA), which is a class of evolutionary algorithms and is inspired by social learning in the society [50]. It has two phases of information: the Population Space, keeping a set of individual solutions, and the Belief Space, keeping various knowledge (e.g., Normative, Situational, Historic, etc.) collected from the population. The two phases communicate through Accept and Influence functions. The Accept function is to permit a selected population to the Belief space, which extracts the knowledge from this population. The Influence function creates new individuals by applying the obtained knowledge.

6.4.1 Initial Population

In the setting of Cultural Algorithms, an initialization method generates individuals randomly. The individuals or chromosomes are solutions to the team formation problem. The individual represents the teams of experts who qualifies to perform the

list of projects, as shown in Figure. 6.3. In our framework, we define the level of expertise of the skill required to complete a task or project. At the same time, we already set the expertise level of experts, as explained in the previous section. When we assign the experts to the required skill of a project, we randomly choose from the qualified list of the experts who have expert level more than or equal to the required level by the project.

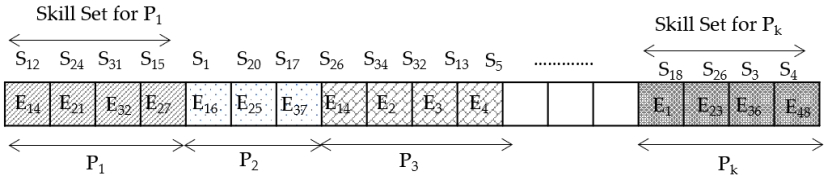


Figure 6.3: The representation of an individual.

6.4.2 Objective Functions

In the spirit of the multi-objective optimization paradigm, we have defined four conflicting objectives to consider when forming a team: the communication cost, the expertise, the geographical proximity, and the collective trust. It would automate a unified way to assemble teams. The multi-objective team formation problem in social networks can be defined in the following:

$$\underset{\mathcal{V}' \subseteq \mathcal{V}}{Minimize} \ \{DCC(\mathcal{V}'), GC(\mathcal{V}')\} \tag{6.10}$$

$$\underset{\mathcal{V}' \subseteq \mathcal{V}}{Maximize} \ \{EL(\mathcal{V}'), CT(\mathcal{V}')\} \tag{6.11}$$

$$Such\ that \sum |V(t_i)| = |\mathcal{V}'|$$

6.4.3 Generate Offspring

To generate new individuals or offspring for each generation, we utilize the benefits of CA that uses the information of knowledge and the benefits of genetic operators.

- The Belief Space: To influence an individual in the population with the knowledge sources, we collect the best individual based on each best objective separately. We then breakdown the chromosome into the sub-chromosomes which represent the solution for a single project t_i from the set of projects $\mathcal{T}$. The sub-chromosomes for the same projects are collected as the best team of experts. The normative knowledge information, what is believed to be good areas to search in each dimension, is defined as the transpose matrix of the selected sub-population composed by the selected teams as described in [10]. To update the topological knowledge component, we mutate from the unexplored expert list. We explain the topological knowledge extraction briefly in the following section.

- Genetic Operators: The genetic operations in MOCA involve crossover and mutations, which help to generate offspring. First, the genetic operator, crossover, or mutation was selected. Following this, the individuals or chromosome is chosen randomly. The chromosome then sends to the ChromosomeBreaker() function to break down into a team of experts per project $\{t_1, t_2, \ldots, t_k\}$. One-point crossover is then applied on each sub chromosomes, as shown in Figure. 6.4.

  Similarly, the mutation operator begins with ChromosomeBreaker() function. It will then mutate an expert from a randomly selected skill of each project. This random expert will be chosen from an unexplored qualified expert list, which is different from qualified experts to a set of experts in the current population. After every genetic operator, sub chromosomes are combined using ChromosomeCombiner() function and then evaluate the objective values of the new chromosome.

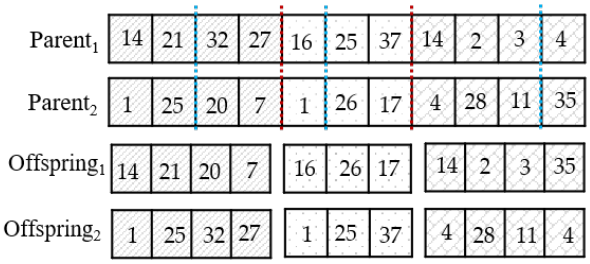


Figure 6.4: The process of crossover, where red line indicates the breaking process from ChromosomeBreaker() function and blue line indicates the random one point crossover on sub chromosome.

6.4.4 Non-Dominated Sorting

For a multi-objective problem, MOCA uses the widely used fast non-dominated sorting approach to compare solutions. We use the ranking procedure as described in [51]. Objective values of each solution compared with every other solution in the population to classify whether are they dominated or non-dominated. All individuals in the first Pareto front are ranked to 1. To find the individuals in the next Pareto front, the first front is removed temporarily, and the above procedure is repeated until they reach the termination condition. In order to find the valid Pareto front, the objective functions are considered collectively when knowledge information influence individuals in the belief space.

6.5 Experiments

This section is to evaluate and analyze the solution produced by MOCA for TFP. To have a unified framework for TFP, we considered highly preferred parameters to discover teams. To the best of our knowledge, there is no real-world dataset that matches entirely to the proposed unified team formation model. Therefore, we examine our experiments on the synthetic dataset that generated similar to real-world collaboration networks. However, we can still examine existing real-world dataset such as DBLP and arXiv by turning off the objectives which are absent on any real

dataset.

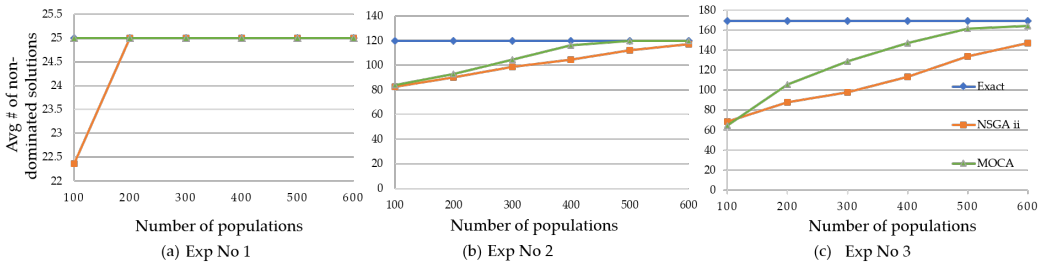


Figure 6.5: Average non-dominated sorting in different number of populations.(a),(b) and (c) are first three instances from table 6.3.

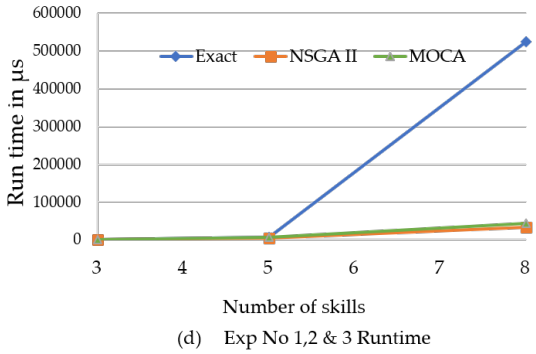


Figure 6.6: The time taken in milli-seconds to find non-dominated sorting.

6.5.1 Dataset

The synthetic social network is generated from LFR benchmark [52] with 200 nodes and 10136 edges. We consider the nodes as experts and edges are the past collaborations. We assign random years from 2010 to 2019 to each collaboration because it requires to evaluate the dynamic nature of communication cost. We collect 50 skill sets as a string array. We then assign a random range of expert lists to each skill with their mastery level score. The emotional index and geographical proximity were generated randomly between 0 and 1. Direct trust value is assigned randomly, either 1 or -1. If any experts never collaborate in the past, we assign 0. To generate the combined trust score of three components: emotional index, direct trust, and skill

similarity, we use the Trade-off solutions. In which, since both direct trust and emotional index are random values, we give less important in our model. Therefore, we assign $\alpha_1 = 0.25, \alpha_2 = 0.5$ and $\alpha_3 = 0.25$.

To evaluate the communication cost, we first generate a graph file that has the detailed list of the connection between each expert and the weight of the connection. Two experts are connected in the network if they interacted with each other, at least in two projects. The weights on edges are computed as:

$$W_{v_i,v_j} = 1 - \frac{|T_{v_i} \cap T_{v_j}|}{|T_{v_i} \cup T_{v_j}|} \tag{6.12}$$

where T_{v_i}(resp.T_{v_j}) is the set of collaboration by V_i (resp. V_j).

The shortest path distance between two experts is computed by an efficient indexing method called 2-hop cover [53]. This indexing technique returns the value of the shortest path between any pair of experts in graphs with any large number of nodes almost instantly.

Experiments	**No of Projects**	**No of Skills**
No 1	1	3
No 2	1	5
No 3	1	8
No 4	3	{2,3,5}
No 5	3	{4,5,6}
No 6	4	{2,3,4,5}
No 7	4	{3,4,5,6}
No 8	4	{5,4,8,6}

Table 6.3: Set of experiments instances where P represents the number of tasks and S represents numbers of skills for each task

6.5.2 Experimental setup and results

The MOCA for TFP is implemented in Java (1.8), and the experiments were performed on an Intel(R) Core(TM)CPU (64 bits), Windows 10 machine with 16 GB of memory.

The configuration details for MOCA are set to the following values, belief space probability = 0.8, crossover probability = 0.16 and mutation probability = 0.04. We tested our experiments by changing the number of population and generation. To compare our model, we implement the exhaustive search algorithm (or exact algorithm) and NSGA II. The exhaustive search algorithm iterates through the entire search space to generate every possible combination of teams. The NSGAII finds the Pareto-optimal team and NSGA II, which is a basic fast non-dominated searching algorithm by Deb at.al. [54]. We can either set the skills and required expert-level manually or generate them randomly.

We create a benchmark table, as shown in table 6.3, to do our experiments. We test our framework with each instance of the table. First, we run each instance with the various number of populations from 100 to 600 by varying the number of iterations from 10 to 60. To find the ideal parameters, we manually set the required skills and expertise level for the project: *"MachineLearning", "Python", "SocialNetworks", "ArtificialIntelligence", "Statistics", "ProjectManagement", "BigData", "DataMining"* and $\{0.7, 0.7, 0.7, 0.5, 0.7, 0.5, 0.7, 0.5\}$. MOCA starts to converge toward the solution earlier than NSGA II in terms of both the iteration and the number of population. As shown in Figure. 6.5, our model finds most of the non-dominated solution with 500 populations and for 60 iterations. Due to time and memory constraints, and the NP-Hard nature of the problem, we are unable to conduct the exhaustive algorithms for complex datasets. However, the initial experiments gave ideal values for populations and iteration because MOCA almost finds the approximate number of non-dominated solutions. The average non-dominated sorting is calculated by running every instance for 5 times. The running time for each instance shows the average milli-seconds in Figure. 6.5 (d).

To check the effectiveness of the expertise level, we observe the solutions with and without the expertise level of the project requirement. For the first case, our

framework will always search within the qualified expert list who satisfies the required skill as well as a level of expertise needed while in the second case, the search space will be the list of experts who satisfy the required skill. The average non-dominated sorting gave a totally different set of teams for both cases. At the same time, if the expertise level of any required skill is not satisfy with the available expert's list, our model will not form any group until it meets the expertise level.

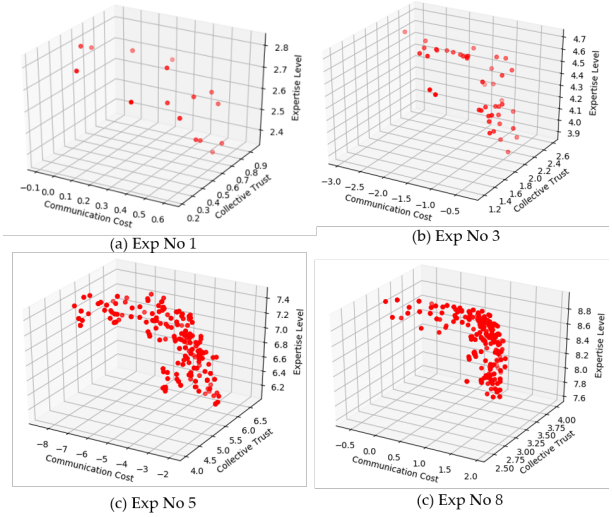


Figure 6.7: Pareto non-dominated fronts from multiple test instances when we consider 3 objectives: Communication Cost, Expertise Level, and Collective Trust.

Any given solution $\mathcal{V}'$ can be evaluated through multiple criteria, such as shortest path distance, diameter distance, minimum spanning tree, and combination of various objectives. We then run our model by turning off some objectives and observe the non-dominated solutions. We randomly plot a few instances from our benchmark to visualize the Pareto front. The Figure. 6.7 and 6.8. We compare MOCA solutions with those obtained by NSGAII and exhaustive algorithms over various criteria.

- DCC($\mathcal{V}'$) $\downarrow$- Dynamic shortest path distance
- CC-Dia($\mathcal{V}'$) $\downarrow$ - Diameter distance
- CT($\mathcal{V}'$) $\uparrow$- Trust value of the team

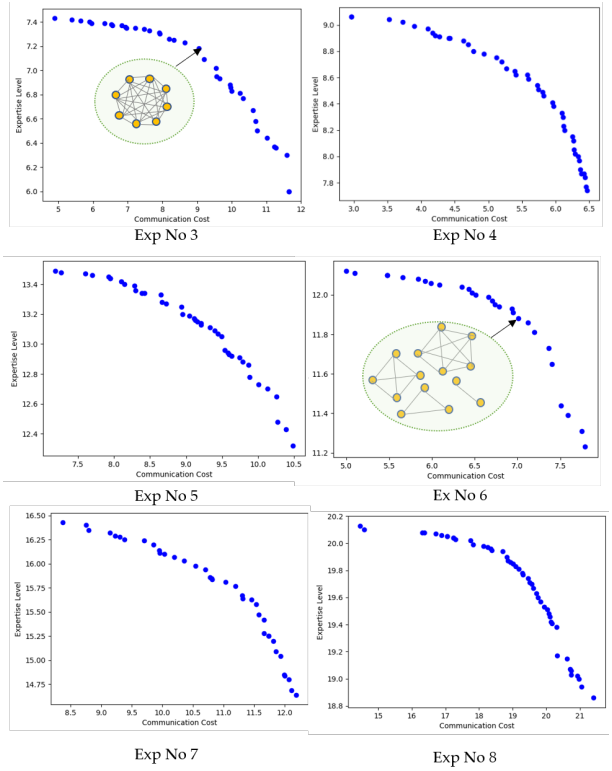


Figure 6.8: Pareto non-dominated fronts from multiple test instances when we consider 2 objectives: Communication Cost and Expertise Level and the topology of some of benchmark's instances

- $\mathrm{EL}(\mathcal{V}')\uparrow$- Expertise level of the team
- $\mathrm{GD}(\mathcal{V}')\downarrow$- Geological proximity of the team

Table 6.4, show a comparison of the values of the solutions evaluated over various criteria from each algorithm. Each column represents the above-listed criteria. The bold value represents the best result from a certain experiment. Each experiment conducted with 500 number of populations and 50 number of iteration. When the number of skills and the number of projects increases, we couldn't perform exhaustive algorithms. At the same time, our model outperforms NSGA II.

Algorithms	**DCC($\mathcal{V}'$)**	**C-Dia($\mathcal{V}'$)**	**GD($\mathcal{V}'$)**	**CT($\mathcal{V}'$)**	**EL($\mathcal{V}'$)**
No 1					
MOCA	**0.3602**	**0.1493**	**2.5932**	**0.8601**	**2.7613**
NSGA II	**0.3602**	**0.1493**	**2.5932**	**0.8601**	**2.7613**
Exhaustive	**0.3602**	**0.1493**	**2.5932**	**0.8601**	**2.7613**
No 2					
MOCA	**1.4894**	**0.2479**	**7.0160**	**2.3441**	**4.5312**
NSGA II	**1.4894**	**0.2479**	**7.0160**	**2.3441**	**4.5312**
Exhaustive	**1.4894**	**0.2479**	**7.0160**	**2.3441**	**4.5312**
No3					
MOCA	**1.2918**	**0.1116**	**21.6713**	**6.8821**	**7.2634**
NSGA II	2.8179	0.2974	26.4526	8.6448	**7.2634**
Exhaustive	**1.2918**	**0.1116**	**21.6713**	**6.8821**	**7.2634**
No 4					
MOCA	**1.3388**	**0.3157**	**11.8213**	**3.9831**	**9.2343**
NSGA II	1.4842	0.4592	11.2794	3.0438	8.5809
Exhaustive	n/a	n/a	n/a	n/a	n/a
No 5					
MOCA	**2.4063**	**0.4614**	**24.5103**	**7.8237**	**13.1715**
NSGA II	3.5522	0.5539	21.7146	7.7308	12.3911
Exhaustive	n/a	n/a	n/a	n/a	n/a
No 6					
MOCA	**1.1418**	**0.2694**	**16.8631**	**5.6907**	**12.9526**
NSGA II	2.0421	0.5787	14.8037	5.0422	11.5140
Exhaustive	n/a	n/a	n/a	n/a	n/a
No 7					
MOCA	**2.4105**	**0.4884**	**27.1238**	**8.8307**	**16.2816**
NSGA II	3.6805	0.7177	25.6325	7.6624	14.9901
Exhaustive	n/a	n/a	n/a	n/a	n/a
No 8					
MOCA	**4.9812**	**0.6106**	**45.3054**	**14.9241**	**20.5816**
NSGA II	7.5	0.9375	36.7541	13.2304	18.8613
Exhaustive	n/a	n/a	n/a	n/a	n/a

Table 6.4: Comparison results of solutions obtained for each experiment instances of table 6.3 over various criteria from MOCA, NSGAII and Exhaustive algorithm.

6.5.3 Run Time

In this section, we discuss the run time of the algorithms in various instances. In experiment No1, the run time of the exhaustive search was very shorter than the other two algorithms. It then increases suddenly, in other instances, from table 6.3. Later we couldn't perform the exact algorithm because of limited memory. As the graph in Figure. 6.5(d), MOCA, and NSGAII run almost to the same time. However, MOCA took a bit more time than NSGAII because MOCA needs to perform extra

operations to extract knowledge and update it to the next generation. In terms of the number of iterations, MOCA begins to converge earlier than NSGAII.

6.6 Conclusions

In this paper, we proposed a unified framework for TFP in social networks using a multi-objective formulation that optimizes the communication cost, team skill mastery level, the collective trust score, and geographical proximity. We introduced two objective functions, the dynamic communication cost, and expertise level that consider the temporal behavior of the social networks. Moreover, we discussed the importance of emotional index in TFP. In addition to this, we evaluated the trust score using various parameters such as EI, profile similarity, and direct trust score. We introduced the new formula for the profile similarity based on similar skills. We solved this problem using the MOCA framework for which normative and topological knowledge are extracted to generate the next population. The experimental evaluation of our method over different tasks, on a synthetic social network graph, showed a diverse set of competitive solutions from four objectives. We generated a benchmark table for the experiments. MOCA was compared with other algorithms, NSGA II, and Exhaustive method. MOCA outperformed NSGA II and deliver solutions more closed to Exhaustive search.

Bibliography

[1] T. Lappas, K. Liu, and E. Terzi, "Finding a team of experts in social networks," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2009, pp. 467–476.

[2] M. Kargar and A. An, "Discovering top-k teams of experts with/without a leader in social networks," in *Proceedings of the 20th ACM international conference on Information and knowledge management*. ACM, 2011, pp. 985–994.

[3] M. Kargar, A. An, and M. Zihayat, "Efficient bi-objective team formation in social networks," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2012, pp. 483–498.

[4] A. Anagnostopoulos, L. Becchetti, C. Castillo, A. Gionis, and S. Leonardi, "Online team formation in social networks," in *Proceedings of the 21st international conference on World Wide Web*. ACM, 2012, pp. 839–848.

[5] F. Farhadi, M. Sorkhi, S. Hashemi, and A. Hamzeh, "An effective expert team formation in social networks based on skill grading," in *2011 IEEE 11th International Conference on Data Mining Workshops*. IEEE, 2011, pp. 366–372.

[6] R. Ponds, F. Van Oort, and K. Frenken, "The geographical and institutional proximity of research collaboration," *Papers in regional science*, vol. 86, no. 3, pp. 423–443, 2007.

[7] J. Juárez and C. A. Brizuela, "A multi-objective formulation of the team formation problem in social networks: preliminary results," in *Proceedings of the Genetic and Evolutionary Computation Conference*. ACM, 2018, pp. 261–268.

[8] Y. Yang and H. Hu, "Team formation with time limit in social networks," in *Proceedings 2013 International Conference on Mechatronic Sciences, Electric Engineering and Computer (MEC)*. IEEE, 2013, pp. 1590–1594.

[9] X. Wang, Z. Zhao, and W. Ng, "A comparative study of team formation in social networks," in *International conference on database systems for advanced applications*. Springer, 2015, pp. 389–404.

[10] K. Selvarajaha, P. M. Zadeha, M. Kargarb, and Z. Kobtia, "Identifying a team of experts in social networks using a cultural algorithm," *Procedia Computer Science*, vol. 151, pp. 477–484, 2019.

[11] C.-T. Li and M.-K. Shan, "Team formation for generalized tasks in expertise social networks," in *Social Computing (SocialCom), 2010 IEEE Second International Conference on*. IEEE, 2010, pp. 9–16.

[12] A. Majumder, S. Datta, and K. Naidu, "Capacitated team formation problem on social networks," in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2012, pp. 1005–1013.

[13] C.-T. Li, M.-K. Shan, and S.-D. Lin, "On team formation with expertise query in collaborative social networks," *Knowledge and Information Systems*, vol. 42, no. 2, pp. 441–463, 2015.

[14] J. Basiri, F. Taghiyareh, and A. Ghorbani, "Collaborative team formation using brain drain optimization: a practical and effective solution," *World Wide Web*, vol. 20, no. 6, pp. 1385–1407, 2017.

[15] W. Chen, J. Yang, and Y. Yu, "Analysis on communication cost and team performance in team formation problem," in *International Conference on Collaborative Computing: Networking, Applications and Worksharing*. Springer, 2017, pp. 435–443.

[16] M. Kargar and A. An, "Teamexp: Top-k team formation in social networks," in *2011 IEEE 11th International Conference on Data Mining Workshops*. IEEE, 2011, pp. 1231–1234.

[17] M. Kargar, M. Zihayat, and A. An, "Finding affordable and collaborative teams from a network of experts," in *Proceedings of the 2013 SIAM International Conference on Data Mining*. SIAM, 2013, pp. 587–595.

[18] C.-T. Li, M.-Y. Huang, and R. Yan, "Team formation with influence maximization for influential event organization on social networks," *World Wide Web*, vol. 21, no. 4, pp. 939–959, 2018.

[19] K. Selvarajah, A. Bhullar, Z. Kobti, and M. Kargar, "Wscan-tfp: weighted scan clustering algorithm for team formation problem in social network," in *The Thirty-First International Flairs Conference*, 2018.

[20] J. H. Gutiérrez, C. A. Astudillo, P. Ballesteros-Pérez, D. Mora-Melià, and A. Candia-Véjar, "The multiple team formation problem using sociometry," *Computers & Operations Research*, vol. 75, pp. 150–162, 2016.

[21] B. Ashenagar, N. F. Eghlidi, A. Afshar, and A. Hamzeh, "Team formation in social networks based on local distance metric," in *2015 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*. IEEE, 2015, pp. 946–952.

[22] K. Selvarajah, P. M. Zadeh, Z. Kobti, M. Kargar, M. T. Ishraque, and K. Pfaff,

"Team formation in community-based palliative care," in *2018 Innovations in Intelligent Systems and Applications (INISTA)*. IEEE, 2018, pp. 1–7.

[23] Y. Han, Y. Wan, L. Chen, G. Xu, and J. Wu, "Exploiting geographical location for team formation in social coding sites," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2017, pp. 499–510.

[24] L. Chen, Y. Ye, A. Zheng, F. Xie, Z. Zheng, and M. R. Lyu, "Incorporating geographical location for team formation in social coding sites," *World Wide Web*, pp. 1–22, 2019.

[25] C. Dorn and S. Dustdar, "Composing near-optimal expert teams: a trade-off between skills and connectivity," in *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*. Springer, 2010, pp. 472–489.

[26] H. Zhu, E. Chen, H. Xiong, H. Cao, and J. Tian, "Ranking user authority with relevant knowledge categories for expert finding," *World Wide Web*, vol. 17, no. 5, pp. 1081–1107, 2014.

[27] M. Neshati, S. H. Hashemi, and H. Beigy, "Expertise finding in bibliographic network: Topic dominance learning approach," *IEEE transactions on cybernetics*, vol. 44, no. 12, pp. 2646–2657, 2014.

[28] J. Zhang, P. S. Yu, and Y. Lv, "Enterprise employee training via project team formation," in *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*. ACM, 2017, pp. 3–12.

[29] A. Bozzon, M. Brambilla, S. Ceri, M. Silvestri, and G. Vesci, "Choosing the right crowd: expert finding in social networks," in *Proceedings of the 16th International Conference on Extending Database Technology*. ACM, 2013, pp. 637–648.

[30] G. K. Awal and K. Bharadwaj, "Team formation in social networks based on collective intelligence–an evolutionary approach," *Applied Intelligence*, vol. 41, no. 2, pp. 627–648, 2014.

[31] T. Largillier and J. Vassileva, "Using collective trust for group formation," in *International Conference on Collaboration and Technology*. Springer, 2012, pp. 137–144.

[32] S. S. Rangapuram, T. Bühler, and M. Hein, "Towards realistic team formation in social networks based on densest subgraphs," in *Proceedings of the 22nd international conference on World Wide Web*. ACM, 2013, pp. 1077–1088.

[33] A. Gajewar and A. Das Sarma, "Multi-skill collaborative teams based on densest subgraphs," in *Proceedings of the 2012 SIAM International Conference on Data Mining*. SIAM, 2012, pp. 165–176.

[34] B. Golshan, T. Lappas, and E. Terzi, "Profit-maximizing cluster hires," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2014, pp. 1196–1205.

[35] W. F. Cascio, "Managing a virtual workplace," *Academy of Management Perspectives*, vol. 14, no. 3, pp. 81–90, 2000.

[36] W. Sherchan, S. Nepal, and C. Paris, "A survey of trust in social networks," *ACM Computing Surveys (CSUR)*, vol. 45, no. 4, p. 47, 2013.

[37] J. Ahn, D. DeAngelis, and S. Barber, "Attitude driven team formation using multi-dimensional trust," in *2007 IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT'07)*. IEEE, 2007, pp. 229–235.

[38] T. Dong-Huynha, N. Jennings, and N. Shadbolt, "Fire: An integrated trust and reputation model for open multi-agent systems," in *ECAI 2004: 16th Euro-*

pean Conference on Artificial Intelligence, August 22-27, 2004, Valencia, Spain: including Prestigious Applicants [sic] of Intelligent Systems (PAIS 2004): proceedings, vol. 110, 2004, p. 18.

[39] J. R. Howells, "Tacit knowledge, innovation and economic geography," *Urban studies*, vol. 39, no. 5-6, pp. 871–884, 2002.

[40] M. Richardson, R. Agrawal, and P. Domingos, "Trust management for the semantic web," in *International semantic Web conference*. Springer, 2003, pp. 351–368.

[41] K. Selvarajah, Z. Kobti, and M. Kargar, "A cultural algorithm for determining similarity values between users in recommender systems," in *International Conference on the Applications of Evolutionary Computation (Part of EvoStar)*. Springer, 2019, pp. 270–283.

[42] J. Zhan and X. Fang, "A computational trust framework for social computing (a position paper for panel discussion on social computing foundations)," in *2010 IEEE Second International Conference on Social Computing*. IEEE, 2010, pp. 264–269.

[43] P. F. Lazarsfeld, R. K. Merton *et al.*, "Friendship as a social process: A substantive and methodological analysis," *Freedom and control in modern society*, vol. 18, no. 1, pp. 18–66, 1954.

[44] M. McPherson, L. Smith-Lovin, and J. M. Cook, "Birds of a feather: Homophily in social networks," *Annual review of sociology*, vol. 27, no. 1, pp. 415–444, 2001.

[45] C.-N. Ziegler and J. Golbeck, "Investigating interactions of trust and interest similarity," *Decision support systems*, vol. 43, no. 2, pp. 460–475, 2007.

[46] M. Pavlov and R. Ichise, "Finding experts by link prediction in co-authorship networks." *FEWS*, vol. 290, pp. 42–55, 2007.

[47] C. Lee and C.-S. Wong, "The effect of team emotional intelligence on team process and effectiveness," *Journal of Management & Organization*, pp. 1–16, 2017.

[48] R. K. Cooper, "Applying emotional intelligence in the workplace," *Training & development*, vol. 51, no. 12, pp. 31–39, 1997.

[49] M. V. Kosti, R. Feldt, and L. Angelis, "Personality, emotional intelligence and work preferences in software engineering: An empirical study," *Information and Software Technology*, vol. 56, no. 8, pp. 973–990, 2014.

[50] R. G. Reynolds, "An introduction to cultural algorithms," in *Proceedings of the third annual conference on evolutionary programming*. World Scientific, 1994, pp. 131–139.

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

[52] A. Lancichinetti, S. Fortunato, and F. Radicchi, "Benchmark graphs for testing community detection algorithms," *Physical review E*, vol. 78, no. 4, p. 046110, 2008.

[53] E. Cohen, E. Halperin, H. Kaplan, and U. Zwick, "Reachability and distance queries via 2-hop labels," *SIAM Journal on Computing*, vol. 32, no. 5, pp. 1338–1355, 2003.

[54] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, "A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: Nsga-ii," in *Interna-*

tional conference on parallel problem solving from nature. Springer, 2000, pp. 849–858.

Chapter 7

Link Prediction by Analyzing Temporal Behavior of Vertices

In the previous chapters, we discussed the problem of forming teams of experts in social networks. We aim to discover teams by optimizing cost functions. We then, in this chapter and chapter 8, focus on predicting future collaboration with the existing teams, which is typically a real-time application of Link Prediction. The link prediction problem focuses on whether a link occurs between any two members of the network or not. So, it is technically a classification problem.

This chapter introduces a method for predicting links in the future by analyzing the temporal behavior of vertices. We tried the evolutionary method to predict the links. The results confirm that it is not a good idea to use the evolutionary method in classification problems. By observing recent researches in classification problems, deep learning methods shows the most accurate solutions. Therefore, we apply a deep learning framework, Multi Layer Perceptron (MLP), to train and test our model.

7.1 Introduction

Social Networks (SN) can be used to model a comprehensive range of real-life phenomena and examine the world around us. It ranges from online social interaction, including Facebook, Twitter, and LinkedIn to human interactions such as co-authorship, healthcare, and terrorist networks. Social networks analysis is the study of such networks to discover common structural patterns and explains their emergence through computational models of network formation. The complexity and dynamics are essential properties of real-world social networks. Since these networks evolve quickly over time through the appearance or disappearance of new links and nodes, the connection becomes stronger and weaker, and underlying network structure changes with time. Therefore it has become high challenges for researchers to examine various research issues in social network analysis such as classification of a node, detecting the communities, formation of teams, and predicting links between nodes.

Understanding the mechanism of how the networks change over time is a crucial problem that is still not well understood [1]. Significant efforts have been made to explain the evolution of networks during the past decades [2]. However, such researches are yet to achieve the desired results, leaving the door open for further advances in the field. Throughout the last decades, analyzing temporal networks has received much attention among researchers as it has enormous applications in different disciplines such as co-authorship [3], the recommendation of friends [4] and website links. Recently, dynamic link predictions have been approached by various mechanisms and achieved promising results. However, the features of networks vary from each other, and the existing studies are not efficient to represent the importance of nodes and links. The objective of this paper is to address these issues and examines the dynamic nature of social networks.

Link prediction problem needs to be solved by determining the potentialities of

the appearance or disappearance of links between all node pairs of the given network [5]. However, for example, in a collaborative network, if two experts (*i.e.*, vertices) collaborate once on any project, their link remains permanent, although any one of them stops interacting with the others. Therefore, their link become weak in the future, while experts who have frequent interactions their link become strong. In this regard, we observe the behavior of individuals in the collaborative network when they need to decide which new collaborations might prove fruitful in addition to existing connections. Before anyone connects with the others in the network, they usually examine several factors, including whether the people are active throughout the past or not, and are they working on similar projects that they have skills. Therefore, our study takes these factors into account and propose a new model, LATB, to predict the links which occur with others in the future. In this paper, we propose a model for link prediction problem on dynamic social networks and make the following major contributions.

1. Active individuals in social networks are popular among both existing members and new members who like to join the network. They believe that active individuals, for instance, in the co-authorship network, always update their research with current trends as well as being open to new ideas. So, to evaluate the activeness of any member, we consider two factors on the temporal network. (a) The score for constructing new connections (b) The score for the increased number of interactions with existing connections (How much the existing link becomes strong). We introduce a new score function to incorporate the impact of the timestamps and the gap between the current time and the time of the interaction occurred. Besides this, we introduce a probability function based on the activeness score of a pair of nodes to decide the likelihood of occurring a new link.

2. The smaller distance between any two individuals is higher the chances of future

interaction. We incorporate the weighted shortest distance in LATB. In addition to this, we include another objective function, the weighted common neighbor index, which incorporates the time to evaluate the changes of strength of the neighbors' relationship.

3. In LATB, we used Multi-Layer Perceptron (MLP) as a classifier to predict the link formation in the future and defined our model as a binary classification problem.

The remaining of the paper is organized as follows. Section 2 describes related works. Section 3 specifies the problem definitions. Section 4 presents the experimental setup and the corresponding results. Finally, Section 6 concludes the research idea of this paper with directions for future work.

7.2 Related Works

Link prediction problem on the static network examines a single snapshot of a network structure at time t as an input, and then predicts possible unobserved links at time $t'(t \leq t')$ [1]. On the other hand, link prediction in dynamic networks investigates the evolution of networks over time as a sequence of snapshots and then predicts new links in the future. This section presents an overview of the link prediction problems on social networks. Several methods have been proposed to deal with the link prediction problem on the temporal network systems during the past decade.

The researchers designed a lot of topology-based similarity metrics for link prediction such as Common Neighbors (CN) [6], Adamic-Adar Coefficient (AA) [7], and Katz (KZ)[8]. Since the weights of links are rarely taken into account, many researchers modified those metrics in order to adopt the dynamic features of the social networks. The authors [9] examine the link prediction based on connection weight score structural properties of a given network. Zhu *et al.* [10] proposed a weighted

mutual information model which is to estimate the effect of network structures on the connection likelihood by considering the benefits of both structural properties and link weights.

Potgieter *et al.* [11] showed that temporal metrics are valuable features in terms of accuracy. Tylenda *et al.* [12] proposed a graph-based link prediction algorithm and integrated it with temporal information and extended the local probabilistic model to involve time awareness. Yao *et al.* [13] used time-decay to manage the weight of the links and modified the common neighbor index to includes nodes in 2-hop. The authors [14] presented a time frame based unsupervised link prediction method for directed and weighted networks and derived a score for potential links in a time-weighted manner.

Tong W *et al.* [15] examined the concepts of the temporal trend of nodes by considering the changes of the degree over time using the structural perturbation method. Munasinge *et al.* [16] studied the impact of a relationship between timestamps of interactions and strength of the link for the future.

Xiaoyi Li *et al.* [17] proposed a deep learning method, conditional temporal restricted Boltzmann machine, which adopted a combination of feature engineering and CNN to predicts links. Recently, Goyal *et al.* [18] proposed DynGEM, which uses the recent advances in deep learning methods, autoencoders for graph embeddings to handle growing, dynamic graphs and for link prediction. Wang *et al.* [19] examined relational deep learning to jointly model high-dimensional node attributes and link structures with layers of latent variables and proposed generalized variational inference algorithm for learning the variables and predicting the links.

7.3 A Model for Dynamic Link Prediction

7.3.1 Problem Definition

A dynamic network is evolving over time and can be considered as a sequence of network snapshots within a time interval. The size of the network can occasionally shrink or expand as the network evolves. In this work, we focus on undirected weighted graphs.

Given a series of snapshots $\{\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_{t-1}\}$ of an evolving graph $\mathcal{G}_T = \langle \mathcal{V}, \mathcal{E}_T \rangle$, where the edge $e = (u, v) \in \mathcal{E}_{t'}$ represents a link between $u \in \mathcal{V}_{t'}$ and $v \in \mathcal{V}_{t'}$ at a particular time t'. The dynamic link prediction approaches attempt to predict the likelihoods of links in the next time step $\mathcal{G}_t$. The list of graphs $\{\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_{t-1}\}$ corresponding to a list of symmetric adjacency matrices $\{A_1, A_2, \ldots, A_{t-1}\}$. The adjacency matrix A_T of $\mathcal{G}_T$ is a $\mathcal{N} \times \mathcal{N}$ matrix where each element $A_T(i, j)$ takes 1 if the nodes $v_i \in \mathcal{V}$ and $v_j \in \mathcal{V}$, are connected at least once within time period T and takes 0 if they are not. Given a sequence of $(t-1)$ snapshots $\{A_1, A_2, \ldots, A_{t-1}\}$, the goal is to predict the adjacency matrix A_t at future time t.

7.3.2 Node Activeness

The idea of node activeness is highly related to the temporal behaviors of nodes. We can determine the active nodes through the analysis of the time-varying historical information of the nodes. To decide the activeness of the nodes, we can examine how they interact with others (nodes) throughout the timeframe. With any temporal network involving humans, we believe that the following factors are highly relevant to decide the activeness of nodes.

New Connections:

A node can remain inactive or active. It can be decided based on how often a node made the new interactions throughout the time frame. Let us consider any two members A and B of a given dynamic network $\mathcal{G}$ with the same number of new connections in the past; A might be connected at an early stage and not creating any new connection later (can be named as sleeping node), while node B formed most of his or her connection at the later stage (active node). The node B would attract more new people and have a high probability of generating new connections in the near future. We believe that considering this behavior of the node is significant in predicting the new links.

Definition 7.1. ***(Score for New Connections)*** *Given a network $\mathcal{G}_T\langle\mathcal{V},\mathcal{E}_T\rangle$, a set of nodes $A = \langle a_1, a_2, \ldots, a_n\rangle$ at time t_m. Let's say the time windows to estimate the new connections are $|t_m^1 - t_m^2|$,$|t_m^2 - t_m^3|$ $\ldots |t_m^{n-1} - t_m^n|$, where $t_m^1, t_m^2 \ldots t_m^n$ are consecutive time stamps. The score of building new connection at time t_m^n of a member or node of the network a_k can be defined as:*

$$\mathcal{SN}(a_k) = \sum_{i=1}^{t_m^n} \frac{\mathcal{NC}_{a_k}^{t_i}}{|t_m^n - t_i| + 1} \tag{7.1}$$

where $\mathcal{NC}_{a_k}^{t_i}$ is the number of new connection made by the node a_k at time t_i. The term $|t_m^n - t_i|$ is the timestamp between current time and the selected time that the number of new connections has been made by the node a_k. If the difference between t_m^n and t_i is high, the value of $\mathcal{NC}_{a_k}^{t_i}$ over $|t_m^n - t_i| + 1$ become smaller although a node made more number of new connections at early time ($\mathcal{NC}_{a_k}^{t_i}$ is large). However, this is opposite for the nodes which made new connection recently. The timestamp has an addition of one to avoid the denominator become infinite when the two timestamps are equal.

The Frequency of Interaction:

In addition to the new interactions, a node can continuously associate with the existing connected nodes. It is another way to decide the activeness of the node. Let us consider two nodes A and B with the same number of existing connections in the past; A might have several interactions at an early stage and not interacting with them later, while B is having frequent interactions with existing connections throughout the time frame. The later one (B) would attract more new nodes and have a high probability of generating new connections in the near future.

Definition 7.2. ***(Score for Frequent Interactions)*** *The score of frequent collaborations with existing connection at time t_m^n of a node a_k can be defined as:*

$$\mathcal{SE}(a_k) = \sum_{i=1}^{t_m^n} \frac{\mathcal{EC}_{a_k}^{t_i}}{|t_m^n - t_i| + 1} \tag{7.2}$$

where $\mathcal{EC}_{a_k}^{t_i}$ is the number of frequent collaboration with existing connection by a node a_k at time t_i.

For instance, let's consider a network of authors (nodes) at certain year (2012) and the number new connection which both A and C made through last seven years (till 2019) can be shown as Figure. 7.1.

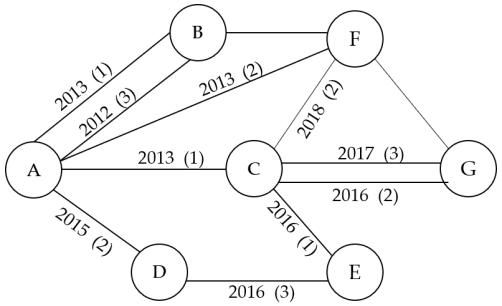


Figure 7.1: Interaction with new and existing nodes throughout the time

$$\begin{aligned}\mathcal{SN}(A) &= \sum_{i=2012}^{2019} \frac{\mathcal{NC}_A^i}{|2019-i|+1} \\ &= \frac{3}{|2019-2012|+1} + \frac{4}{|2019-2013|+1} + \frac{2}{|2019-2015|+1} \\ &= 1.382\end{aligned}$$

Similarly, the value of $\mathcal{SN}(C)$ can be evaluated to 2.89 as the information given in Figure. 7.1. Although both A and C made the equal number of new connections ($= 9$) in the past seven years, the score has a huge difference. The reason is C built new connections recently than A. This shows that nodes, which generate new connections recently have more attraction than others. Likewise, we can evaluate the score value of active nodes in terms of frequent interaction with existing connections.

Since both building new interactions and frequent interactions with existing connections have an influence on deciding the active node, the combination of these scores is a proper way to determine the score of the active node as given in equation 7.3.

$$\mathcal{SA}(a_k) = \lambda \sum_{i=1}^{t_m^n} \frac{\mathcal{NC}_{a_k}^{t_i}}{|t_m^n - t_i| + 1} + (1-\lambda) \sum_{i=1}^{t_m^n} \frac{\mathcal{EC}_{a_k}^{t_i}}{|t_m^n - t_i| + 1} \tag{7.3}$$

where λ is a tradeoff value between the score for building new connections and expanding existing connections. Algorithm 3 describe the step by step process of calculating node activeness score for all nodes and store it in a lookup table.

At this point, every node is assigned by a score based on their activity on dynamic networks. However, to maintain the range of values between 0 and 1, we normalize each score. Inspired by configuration model, the probability $\mathcal{P}_{a_i,a_j}$ of a link exists between any two nodes a_i and a_j would be proportional to $\mathcal{SA}(a_i).\mathcal{SA}(a_j)$. Since the probability should be between 0 and 1, we drive the equation for this value by multiplying the reciprocal of the total activeness of the nodes in the networks as given in equation 7.5.

$$\mathcal{P}_{a_i,a_j} = \frac{\mathcal{SA}(a_i)\mathcal{SA}(a_j)}{\sum_{k=1}^{n} \mathcal{SA}(a_k)} \tag{7.4}$$

Algorithm 3 Node Activeness

Input: Current Time stamp t_c, Snapshots of a given network G_T, List of vertices L_V, Trade off value λ
Output: Activeness Score lookup table $\mathcal{SA}$

$\mathcal{SA} \leftarrow \{\}$
$T[\,] \leftarrow \{t_1, t_2, \ldots t_k\}$ time steps
for all $n \in L_V$ **do**
  $Ex \leftarrow 0$ Existing Connection Score
  $Nw \leftarrow 0$ New Connection Score
  $nb \leftarrow \Gamma\,[G_{t_0}(n)]$ collaborated nodes at time $t = 0$
  for all t_i in T **do**
    $nb_i[\,] \leftarrow \Gamma\,[G_{t_i}(n)]$
    for all $m \in nb$ **do**
      if $m \in nb_0$ **then**
        $Ex \leftarrow Ex + \frac{w_{G_{t_i}}(m,n)}{t_c - t_i + 1}$
      else
        $Nw \leftarrow Nw + \frac{w_{G_{t_i}}(m,n)}{t_c - t_i + 1}$
      end if
    end for
    $nb \leftarrow nb \cup nb_i$
  end for
  $P \leftarrow \lambda.Ex + (1 - \lambda)Nw$
  $\mathcal{SA} \leftarrow \{\mathrm{n}\colon P\}$
end for
return $\mathcal{SA}$

where $\mathcal{P}_{a_i,a_j}$ is the probability of existing link between node a_i and a_j, $\mathcal{SA}(a_i)$ and $\mathcal{SA}(a_j)$ are the popularity scores of nodes a_i and a_j respectively and n is the total number of nodes in the networks.

We name our proposed method LATB (Link prediction by Analyzing Temporal Behaviour of vertices), because it highlights the behaviors of vertices.

7.3.3 Similarity Metrics

In the past, the majority of researches [1, 20] have examined the accuracy of several heuristics for link prediction such as Adamic Adar, Preferential Attachment, and Jaccard Coefficient. In this research, we consider two modified forms of heuristics:

weighted shortest path distance and weighted common neighbors.

Weighted Shortest Path Distance:

In the undirected social network G, if some nodes have past interaction, their associated nodes in G are connected by an edge. If many levels of past interactions between two nodes are taken into account, then the input graph G is weighted. In this case, the smaller the edge weight between two nodes, the two nodes had more interactions in the past and have higher chances of interactions in the future. The distance between two nodes a_i and a_j, specified as $dist(a_i, a_j)$, is equal to the sum of the weights on the shortest path between them in the input graph G. If a_i and a_j are not connected in graph G,i.e., there is no path between a_i and a_j in G, the distance between them is set to ∞.

$$\mathcal{SD}_{a_i,a_j} = wdist(a_i, a_j) \tag{7.5}$$

Weighted Common Neighbors:

The Common Neighbors (CN) is the most widely used index in link prediction and evidence to the network transitivity property. It counts the number of common neighbors between node pair a_i and a_j. Newman *et al.* [6] has estimated this quantity in the context of collaboration networks. The probability that a_i and a_j collaborate in the future can be written as 7.6.

$$\mathcal{CN}_{a_i,a_j} = |\Gamma(a_i) \cup \Gamma(a_j)| \tag{7.6}$$

where $\Gamma(a_i)$ and $\Gamma(a_j)$ consists of number of neighbors of the node a_i and a_j in $\mathcal{G}$ respectively.

As mentioned in the above definition, the common neighbors only consider the binary relations between nodes and ignore the time-varying nature and number of link occurrences. We adopt the time-varied weights into the common neighbors, which

can give better predictions [21].

$$\mathcal{CN}^{tw}_{a_i,a_j} = \sum_{|\Gamma(a_i)\cup\Gamma(a_j)|} \mathcal{W}^t(a_i,a_k) + \mathcal{W}^t(a_k,a_j) \tag{7.7}$$

where $\mathcal{W}^t(a_i,a_k) = \mathcal{W}(a_i,a_k) - \beta(t'-t)$, $\mathcal{W}(a_i,a_k)$ is original weight at time t, β is an attenuation factor and t' is the time considered for prediction.

7.3.4 Multilayer Perceptron (MLP) Framework

We treat the link prediction problem as a binary classification problem. We used MLP as a classifier. In this regards, we generate a dataset for all existing links in a last time step of a given dynamic network $\mathcal{G}_\mathcal{T}$, for the positive link class, where for any two vertices i and j, the link between i and j, $ij \in E_t$. We generate negative link class, where any two vertices i and j, $ij \notin E_t$ by using downsampling technique to avoid imbalance problem. We assign the binary cross-entropy as loss function, which can be written as:

$$BCE = -y.log(p) - (1-y)log(1-p) \tag{7.8}$$

where y is binary indicator (0 or 1), p is predicted probability.

Finally, we build and train a neural network for link prediction. MLP is one of the most common and a variant of the original Perceptron model [22]. Here, we only briefly discuss the components of an MLP since this paper is not about MLP innovations. A typical MLP system can be built with layers of neurons, as shown in Figure. 7.2. Each neuron in a layer calculates the sum of its inputs (x) that are carried through an activation function (f). The output (O) from the network can be written as:

$$\mathcal{O}_{jk} = F_k(\sum_{i=1}^{N_{k-1}} w_{ijk}x_{i(k-1)} + \beta_{jk}) \tag{7.9}$$

where $\mathcal{O}_{jk}$ is the neuron j^{th} output at k^{th} layer and β_{jk} is bias weight for neuron j in

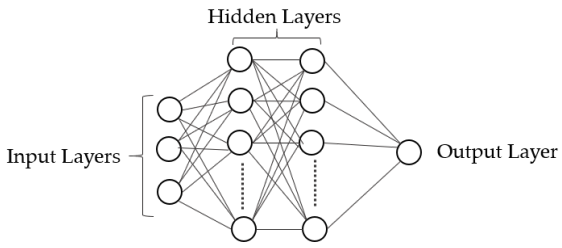


Figure 7.2: MLP neural network

layer k, respectively.

7.4 Experimental Results

We conduct extensive experiments to test our model with five real-world dynamic networks and use AUC (Area Under Curve) as evaluation metrics.

7.4.1 Dataset

We use six real world dynamic networks: **Enron** corpus [23] and **Radoslaw** [24] are email communication networks. Each node specifies an employee and link represents email conversation among employees. Enron has the details from 6 January 1998 until 4 February 2004 while Randoslaw is from January 2^{nd} 2010 to September 30^{th} 2010. **Contact** [25] is data from wireless devices carried by people. Every node is people, and a link established when they contacted. The contact list represents the active contacts during 20-second intervals of the data collection. **College Messages** [26] have private messages sent on an online social network at the University among college people. **EU-core** [27] is an email data from a large European research institution. Link is the communication between members from 4 different departments. **Mathoverflow** [27] has the interactions on the stack exchange web site Math Overflow.

In the beginning, we sort the dataset in ascending order of time, and then we process a sequence of snapshots for each dataset at a fixed interval. We split every

Dataset	$\|V\|$	$\|E_T\|$	Time span in days
Enron	151	50571	165
Radoslaw	167	82900	272
Contact	274	28200	4
CollegeMessages	1899	59835	193
EU-core	986	332334	803
Mathoverflow	24818	506550	2350

Table 7.1: The statistical information of each real-world dynamic networks.

dynamic networks into five time frames G_1, G_2, G_3, G_4, G_5. We evaluate each scoring value at the last snapshot.

7.4.2 Experimental Setup

We implement our model in Python3, processed the dataset on IBM cluster, the specification of POWER8 52 processor 256 GB of RAM. We trained and tested our model in an Nvidia GTX 1050Ti, 4 GB GPU with 768 CUDA cores.

In the weighted common neighbor index, we set the attenuation factor β to 0.001. To evaluate the active score of the nodes, we assign tradeoff factor λ to 0.5, because we believe that both the score for new connections and the score for frequent interaction with existing connections are equally important to decide the activeness of a person.

In the MLP, the first layer has four neurons with the ReLu activation function. We use two hidden layers of 32 neurons. The output layer contains a single neuron with the Sigmoid activation function. We train the neural network for 100 epochs. We use 80% training set, 10% validation set, and 10% testing set. We repeat the above process for ten times and find the average AUC.

7.4.3 Baseline Methods

We use various methods as the baselines, including classical methods such as Common Neighbors (CN), Jaccard coefficient (JC), Adamic Adar (AA) and Preferential attachment (PA), and network embedding methods such as node2vec, LINE, DeepWalk, and SDNE. The brief introduction of these methods is listed as follow:

- Common Neighbors (CN) [6]: It is one of the most common measurements used in link prediction problem. Having a large number of the common neighbors easily create a link.
- Jaccard Coefficient (JC): It is a normalized form of the CN index.
- Adamic-Adar Coefficient (AA) [7]: It evaluates the impotency of a node when having less number of neighbors when predicting links.
- Preferential Attachment (PA) [28]: It generates the belief that nodes with large number of neighbors are more likely to form more in the future.
- node2vec [29]: It is a node embedding method, which learns nodes representation of network by preserving higher-order proximity between nodes. It used a higher probability of node occurrence in a fixed-length random walk.
- LINE [30]: It used an objective function to preserves the first-order and second-order neighborhoods to learn node representations, most similar to node2vec. It is useful to apply for large-scale network embedding.
- DeepWalk [31]: It used random walk model to learn vertex representations. This embedding can be used to predict link existence.
- SDNE [32]: It used both the first-order and second-order proximities together in an autoencoder based deep model to generate the vertex representations.

Regarding the implementations, we evaluated the Link prediction problem from the original code by the authors for node2vec[1], LINE[2], DeepWalk[3] and SDNE[4].

Dataset	CN	JC	AA	PA	node2vec	LINE	DeepWalk	SDNE	LATB
Enron	0.8106	0.8751	0.8970	0.8442	0.7596	0.5042	0.7190	**0.9437**	0.9302
Radoslaw	0.8417	0.8307	0.9028	0.8753	0.7417	0.6153	0.7342	0.8709	**0.9457**
Contact	0.8457	0.9141	0.9142	0.9027	0.8741	0.7360	0.8451	0.9376	**0.9906**
CollegeMessages	0.5742	0.5774	0.5843	0.5901	0.7049	0.4905	0.7506	0.7806	**0.9576**
EU-core	0.9227	0.9302	0.9341	0.7553	0.8602	0.6587	0.8201	0.9574	**0.9626**
Mathoverflow	0.7774	0.7692	0.7430	0.7783	0.7478	0.6587	0.7456	0.9574	**0.9968**

Table 7.2: Experimental Results based on AUC by comparing to Classic and Embedding methods.

7.4.4 Results

Our model achieves a significant improvement compared to other methods in various dynamic networks except the dataset Enron, which is better in SDNE. The standard network embedding methods, node2vec and LINE, perform the worst than other methods in terms of AUC. LATB performs significantly better in Contact and Eu-Core networks above 96% while the classic and embedding methods achieve below 92%. Moreover, LATB has an AUC higher than 93% among all tested dynamic networks. We can conclude that LATB can perform well in both very sparse (Enron) and dense (Radoslaw) networks. Tables 7.2 presents the results comparison of other methods with LATB.

[1]https://github.com/aditya-grover/node2vec
[2]https://github.com/tangjianpku/LINE
[3]https://github.com/phanein/deepwalk
[4]https://github.com/xiaohan2012/sdne-keras

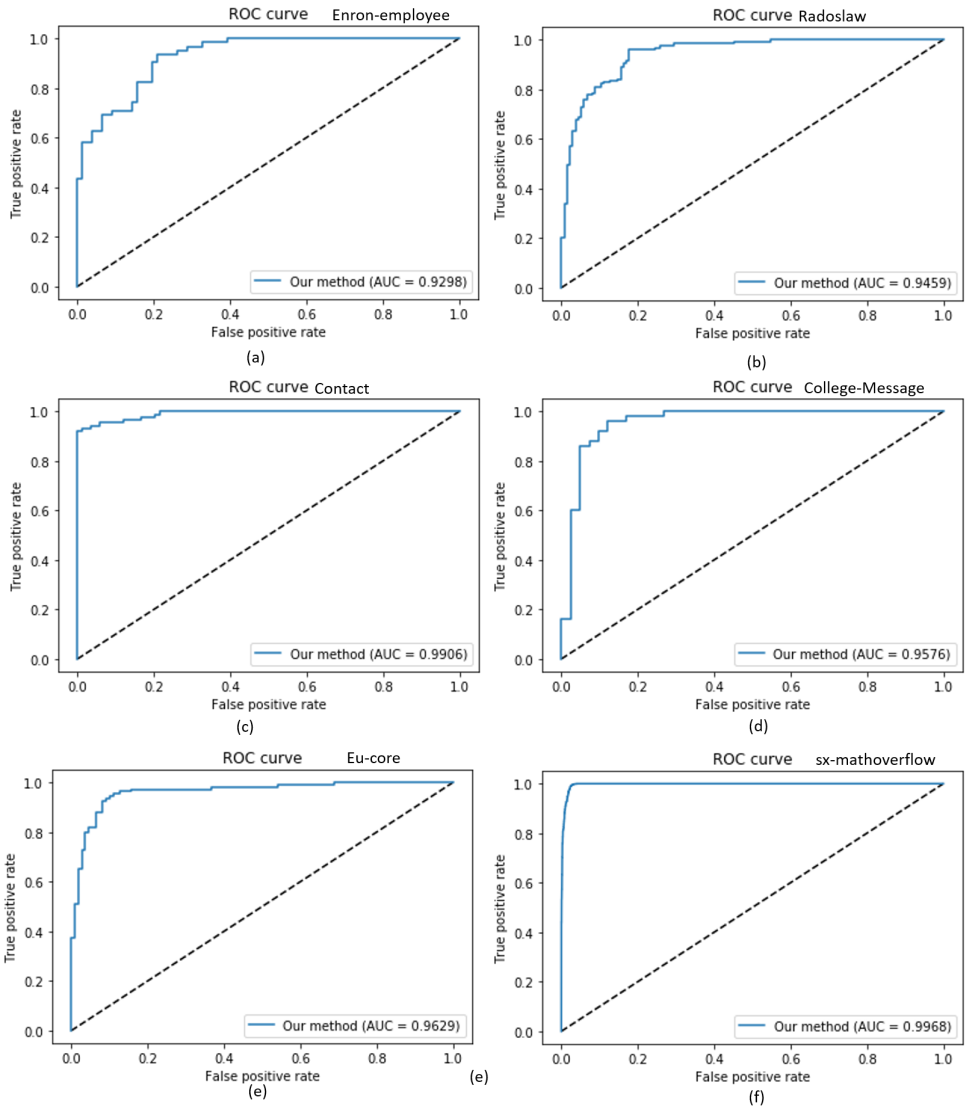


Figure 7.3: Experiments on ROC curve. (a) ROC curve on College Message. (b) ROC curve on Enron-employee. (c) ROC curve on Radoslaw.(d) ROC curve on Contact. (e) ROC curve on Eu-Core.

7.5 Conclusions

In this paper, we propose a model for link prediction in dynamic networks by analyzing temporal behaviors of vertices, named LATB. To model the evolving pattern of each vertex, we propose a new scoring method, which can engage the historical changes of vertices. To further address LATB, temporal changes of vertices are analyzed in two

ways to measure the activeness of a node: how often a vertex interacts with existing connected nodes - to measure the strength of the relationship with its neighbors, and how fast and often collaborate with new nodes. Because both measures have a strong influence on deciding the node activeness, we introduce a probability function based on the activeness of nodes to evaluate the chances of being connected in the future. We also use two other weighted indexes: shortest distance and common neighbors, to incorporate the time-varying nature and number of link occurrences in neighbor nodes. In LATB, MLP is used as a classifier, which results in the status of link existence. Empirically, we compare LATB with classical methods and traditional embedding methods in five different real-world dynamic networks. Overall our model, LATB, achieves significant improvements and reaches above 93% of AUC.

Acknowledgment

This research work was supported by International Business Machines (IBM); experiments were conducted on a high performance IBM Power System S822LC Linux Server.

Bibliography

[1] D. Liben-Nowell and J. Kleinberg, "The link-prediction problem for social networks," *Journal of the American society for information science and technology*, vol. 58, no. 7, pp. 1019–1031, 2007.

[2] S. N. Dorogovtsev and J. F. Mendes, "Evolution of networks," *Advances in physics*, vol. 51, no. 4, pp. 1079–1187, 2002.

[3] S. Aydın, "Link prediction models for recommendation in academic collaboration network of turkey," Ph.D. dissertation, 2017.

[4] S.-H. Yang, A. J. Smola, B. Long, H. Zha, and Y. Chang, "Friend or frenemy?: predicting signed ties in social networks," in *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2012, pp. 555–564.

[5] P. Wang, B. Xu, Y. Wu, and X. Zhou, "Link prediction in social networks: the state-of-the-art," *Science China Information Sciences*, vol. 58, no. 1, pp. 1–38, 2015.

[6] M. E. Newman, "Clustering and preferential attachment in growing networks," *Physical review E*, vol. 64, no. 2, p. 025102, 2001.

[7] L. A. Adamic and E. Adar, "Friends and neighbors on the web," *Social networks*, vol. 25, no. 3, pp. 211–230, 2003.

[8] L. Katz, "A new status index derived from sociometric analysis," *Psychometrika*, vol. 18, no. 1, pp. 39–43, 1953.

[9] T. Murata and S. Moriyasu, "Link prediction of social networks based on weighted proximity measures," in *Proceedings of the IEEE/WIC/ACM international conference on web intelligence*. IEEE Computer Society, 2007, pp. 85–88.

[10] B. Zhu and Y. Xia, "Link prediction in weighted networks: A weighted mutual information model," *PloS one*, vol. 11, no. 2, p. e0148265, 2016.

[11] A. Potgieter, K. A. April, R. J. Cooke, and I. O. Osunmakinde, "Temporality in link prediction: Understanding social complexity," *Emergence: Complexity & Organization (E: CO)*, vol. 11, no. 1, pp. 69–83, 2009.

[12] T. Tylenda, R. Angelova, and S. Bedathur, "Towards time-aware link prediction in evolving social networks," in *Proceedings of the 3rd workshop on social network mining and analysis*. ACM, 2009, p. 9.

[13] Yao, Lin, L. Wang, L. Pan, and K. Yao, "Link prediction based on common-neighbors for dynamic social network," *Procedia Computer Science*, vol. 83, pp. 82–89, 2016.

[14] M. Kaya, M. Jawed, E. Bütün, and R. Alhajj, "Unsupervised link prediction based on time frames in weighted–directed citation networks," in *Trends in Social Network Analysis*. Springer, 2017, pp. 189–205.

[15] T. Wang, X.-S. He, M.-Y. Zhou, and Z.-Q. Fu, "Link prediction in evolving networks based on popularity of nodes," *Scientific reports*, vol. 7, no. 1, p. 7147, 2017.

[16] L. Munasinghe and R. Ichise, "Time aware index for link prediction in social networks," in *International Conference on Data Warehousing and Knowledge Discovery*. Springer, 2011, pp. 342–353.

[17] X. Li, N. Du, H. Li, K. Li, J. Gao, and A. Zhang, "A deep learning approach to link prediction in dynamic networks," in *Proceedings of the 2014 SIAM International Conference on Data Mining*. SIAM, 2014, pp. 289–297.

[18] P. Goyal, N. Kamra, X. He, and Y. Liu, "Dyngem: Deep embedding method for dynamic graphs," *arXiv preprint arXiv:1805.11273*, 2018.

[19] H. Wang, X. Shi, and D.-Y. Yeung, "Relational deep learning: A deep latent variable model for link prediction," in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[20] M. Pavlov and R. Ichise, "Finding experts by link prediction in co-authorship networks." *FEWS*, vol. 290, pp. 42–55, 2007.

[21] S. Huang, Y. Tang, F. Tang, and J. Li, "Link prediction based on time-varied weight in co-authorship network," in *Computer Supported Cooperative Work in Design (CSCWD), Proceedings of the 2014 IEEE 18th International Conference on*. IEEE, 2014, pp. 706–709.

[22] F. Rosenblatt, *The perceptron: A theory of statistical separability in cognitive systems*. United States Department of Commerce, 1958.

[23] B. Klimt and Y. Yang, "The enron corpus: A new dataset for email classification research," in *European Conference on Machine Learning*. Springer, 2004, pp. 217–226.

[24] R. Rossi and N. Ahmed, "The network data repository with interactive graph

analytics and visualization," in *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[25] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott, "Impact of human mobility on opportunistic forwarding algorithms," *IEEE Transactions on Mobile Computing*, no. 6, pp. 606–620, 2007.

[26] P. Panzarasa, T. Opsahl, and K. M. Carley, "Patterns and dynamics of users' behavior and interaction: Network analysis of an online community," *Journal of the American Society for Information Science and Technology*, vol. 60, no. 5, pp. 911–932, 2009.

[27] A. Paranjape, A. R. Benson, and J. Leskovec, "Motifs in temporal networks," in *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*. ACM, 2017, pp. 601–610.

[28] A.-L. Barabâsi, H. Jeong, Z. Néda, E. Ravasz, A. Schubert, and T. Vicsek, "Evolution of the social network of scientific collaborations," *Physica A: Statistical mechanics and its applications*, vol. 311, no. 3-4, pp. 590–614, 2002.

[29] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2016, pp. 855–864.

[30] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "Line: Large-scale information network embedding," in *Proceedings of the 24th international conference on world wide web*. International World Wide Web Conferences Steering Committee, 2015, pp. 1067–1077.

[31] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2014, pp. 701–710.

[32] D. Wang, P. Cui, and W. Zhu, "Structural deep network embedding," in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, 2016, pp. 1225–1234.

Chapter 8

Dynamic Network Link Prediction by Learning Effective Subgraphs using CNN-LSTM

This chapter is also to predict future collaborations. In previous work, we propose a probability score function based on the activeness of team members and use a given entire network for the prediction process. It leads to an expensive process with large size networks. Motivated by our work in "Link Prediction by Analyzing Common Neighbors Based Subgraphs using Convolutional Neural Network," we focus on the temporal behavior of common neighbor based subgraphs and extract the heuristic features for the link prediction process. The subgraph based method helps us in handling complex networks.

8.1 Introduction

Dynamic network analysis has become an important research problem in recent years because it resembles the evolving nature of real-world networks. It has taken a great deal of attention from various fields, including social science [1], economics [2], and

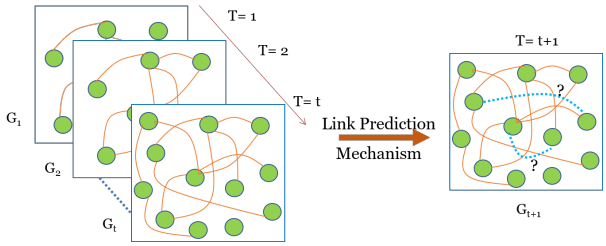


Figure 8.1: The representation of a dynamic network G with the series of snapshots from time 1 to t as a input and a snapshot at time $t+1$ as a output

biology [3]. Dynamic networks evolve over time, and nodes and links may appear or disappear as time goes by. One of the primary areas of research in dynamic networks is temporal link prediction, which attempts to predict the links in the future using the transformation of a sequence of networks. Link Prediction (LP) has several applications including friend recommendation [4], classify the behavior and motion of people [5], and disease gene prediction [6].

Numerous studies have been performed in a static network setting, which considers a single snapshot of a network at time t and is used to determine new links in time $t'(>t)$. Simple heuristic methods, often based on topological properties of the network, such as common neighbors [7], Adamic-Adar [4] and Katz [8] or a combination of such heuristics are well-defined for static networks. Link prediction in a dynamic network is a challenging and complex process. It has a completely new dimension of analysis because the history of network evolution provides more information to detect potential or future links. The dynamic network settings can be generally formulated as the sequence of network snapshots, as shown in Figure 8.1, where the behavior of each snapshot can be described as a static network at a time. To deal with dynamic network link prediction, various methods have been proposed in the literature [9, 10, 11, 12]. These methods include network embedding techniques such as DeepWalk [13], LINE [14] and Node2Vec [12] and deep learning techniques [9, 10, 11]. The approach in [15] and [16] have explored the usage of heuristic methods in the dynamic network link prediction.

Most of the existing approaches in both static and dynamic settings focused only on the target nodes, source and destination of the link and entire network for the prediction. However, the target nodes and their neighbor nodes play a high impact on link prediction, and analyzing the portion of the whole network reduce time complexity. Recent ground-breaking methods in static networks, WLMN [17] and SEAL [18] proposed neural network approaches to automate the selection of best heuristic for a given network, and introduced subgraph extraction methods, based on neighbor nodes, of the target links for the prediction. However, PLACN [19] claimed that subgraphs by common neighbor nodes of target link have additional information than the subgraph from just neighbor nodes, and achieved outstanding results in various types of static networks. Motivated from this, we extract subgraph from common neighbors of target links and extend the benefits of heuristics to the dynamic network settings. We believe that considering subgraph based on common neighbors of a target link bring a huge advantage to analyze the evolving pattern of the target link in the dynamic network.

To the best of our knowledge, we are the first people using the common neighbor based subgraph for the dynamic network link prediction problems. Our proposed model, DLP-LES, begins with extraction of common neighbor based subgraph from the last snapshot of a given dynamic network, and analyzes the transitional patterns of subgraph using heuristic features throughout each time step. Due to complexity, most of the research in dynamic settings ignored the link weight, and only considered the existence and absence of the link. In DLP-LES, we include link weight as an additional information with heuristic features. Besides this, we construct a lookup table with every information of links of a given dynamic network. The primary purpose of the lookup table is to reduce the time and space complexity when we frequently use the information of the same links. We elaborate this further in the section for constructing the lookup table. Thereafter, this study introduces an efficient encoding method to

label the subgraph's nodes. It is another significant task in our model to maintain the consistency of the subgraph when we train the neural networks. We believe that examining the evolving heuristic features of the subgraph has a significant impact on introducing a new link between any two nodes of a dynamic network.

We summarize our main contributions as follows:

- We introduce a method to generate a lookup table to keep the record of links' information of a given dynamic network, and use a hashing method to fetch essential information when required.
- We introduce a novel encoding method for subgraph labeling.
- We propose an algorithm to construct feature matrices for the subgraph efficiently.
- We propose a new framework, DLP-LES, for the dynamic network link prediction using Convolutional Neural Networks (CNN) to extract higher-level features of subgraph efficiently and Long Short-Term Memory (LSTM) neural networks to learn long-range dependencies of sequential data and capture the evolving patterns of the subgraph in the dynamic networks.

8.2 Related Works

Link Prediction in Dynamic Networks (DN) is one of the hot topics in social network analysis. Modeling this problem is a complex and highly challenging process. Diverse methods have been proposed in the literature to improve the accuracy of the predictions.

Heuristic methods such as common neighbors [7], Adamic-Adar [4] and Katz [8] consider the topological structure to predict the links in the future, which are very famous for static networks. Yao *et al.* [20] proposed a modified common neighbors

formula and use of time-decay to handle DN. Some others [16, 15] extended the application of these heuristics to DN settings. Chiu *et al.* [16] proposed a weak estimator to decide the link existence based on a random probability function, while Kaya *et al.* [15] explored aggregate heuristic metrics by weighting snapshots.

Besides heuristic based prediction methods, various machine learning techniques have been applied for LP in DN. Gao *et al.* [21] performed a method by combining the latent matrix factorization method and graph regularization technique to learn the structural information of time evolving patterns of links. Yu *et al.* [22] proposed a model (LINE) with spatial and temporal consistency to tackle DN prediction. They represented the network structure as a function of time. Ma *et al.* [23] proposed a non-negative matrix factorization (NMF) framework which incorporated the dynamic information of historical snapshots by using the graph regularization technique. De *et al.* [24] proposed an extended node2vec method to apply on a dynamic setting.

In addition to the above two methods, deep learning approaches have become cutting-edge techniques in DN link predictions. Li *et al.* [9] proposed a framework using boltzmann machine which predicts links based on individual transition variance in addition to influence introduced by local neighbors. The authors of [25] proposed a network embedding method to handle DN settings. They incorporated both the internal and dynamic transition structures in their design. Lei *et al.* [26] proposed a model using GCN, LSTM, and GAN to solve the challenges in temporal LP. They used graph convolutional network (GCN) to study the local topological structure, LSTM to analyze the evolving features of networks, and generative adversarial networks (GAN) to handle weighted DN.

Some other methods are also used in temporal network LP. CA Bliss *et al.* [27] employed evolutionary algorithms to predict the links on DN by applying the Covariance Matrix Adaptation Evolution Strategy.

8.3 Problem Definition

A dynamic network can be defined as a sequence of network snapshots considered within a specific time interval where as a static network does not change the topological structure over time. In this paper, we consider an undirected, weighted dynamic network.

Given a series of snapshots $\{\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_t\}$ of an evolving graph $\mathcal{G}$, where $\mathcal{G}_p = \langle \mathcal{V}, \mathcal{E}_p \rangle$ represents a snapshot of the given dynamic network at time p. In this study, $\mathcal{V}$ specifies the same vertices shared by all snapshots. $\mathcal{E}_p$ specifies the links or edges of the snapshot at time p. A snapshot $\mathcal{G}_p$ can be treated as a static network, and can be written as an adjacency matrix $A_p = [a_t(i,j)]_{|V|\times|V|}$ to represents the corresponding static topological structure, where $a_p(i,j) > 0$ if the vertices $v_i \in \mathcal{V}$ and $v_j \in \mathcal{V}$ are connected, otherwise, $a_p(i,j) = 0$. The sequence of graphs $\{\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_t\}$ correspond to a list of symmetric adjacency matrices $\{A_1, A_2, \ldots, A_t\}$.

Definition 8.1. ***(Link Prediction in Dynamic Networks)*** *Given that a sequence of snapshots with length k have the corresponding adjacency matrices* $\{A_{t-k}, A_{t-k+1}, \ldots, A_t\}$*, the primary objective of link prediction in dynamic networks is to model a framework to learn the following function to predict the topological changes, mainly in links at time* $t+1$*:*

$$A_{(t+1)} = f(A_{t-k}, A_{t-k+1}, , \ldots, A_t) \tag{8.1}$$

where $f(A_{t-k}, A_{t-k+1}, , \ldots, A_t)$ represents the model required to predict the adjacency matrix $A_{(t+1)}$ at time t.

8.4 Modeling Dynamic Networks Link Prediction

In this section, we discuss the backgrounds of required theories and techniques to model a framework for predicting links in the future.

8.4.1 Heuristic Methods

Several heuristics have been proposed extensively to solve link prediction problems on static networks, such as Common neighbors, Adamic-Adar and Katz. We can categorize them as first, second, and high order heuristics based on their complexity to perform. The first and second order heuristics are efficiently computable, and measure diverse aspects of the network topology such as closeness and similarity between any two nodes in the social networks. The following section lists down five such heuristics used in this paper, where $\Gamma(v)$ and $\Gamma(u)$ specify the set of neighbors for nodes v and u respectively.

Common Neighbors (CN)

The idea of CN is that if the nodes share links with other nodes, the chances of forming a new link is high. It is the most simplest method and counts the number of neighbors that any two vertices v and u directly interact with.

$$\mathcal{CN} = |\Gamma(v) \cap \Gamma(u)| \tag{8.2}$$

Jaccard Coefficient (JC)

The CN measures the relative similarities between any two nodes because it does not consider the proportion of links shared; it is not normalized. JC produces the

normalized form of CN based on the total number of neighbors both v and u have.

$$\mathcal{JC} = \frac{|\Gamma(v) \cap \Gamma(u)|}{|\Gamma(v) \cup \Gamma(u)|} \tag{8.3}$$

Adamic-Adar (AA)

The AA is the modified version of JC. The primary purpose of AA is to give a higher priority to the common neighbors with very few neighbors or lower degree.

$$\mathcal{AA} = \sum_{k \in |\Gamma(v) \cup \Gamma(u)|} \frac{1}{log|\Gamma(k)|} \tag{8.4}$$

Preferential Attachment (PA)

The concept of PA is if a node has a higher degree, the chances of making new connections is high.

$$\mathcal{PA} = |\Gamma(v) * \Gamma(u)| \tag{8.5}$$

Resource Allocation (RA)

RA metric is much more similar to AA. The difference is that RA gives higher priority to low-degree common neighbors than AA.

$$\mathcal{RA} = \sum_{k \in |\Gamma(i) \cup \Gamma(j)|} \frac{1}{|\Gamma(k)|} \tag{8.6}$$

8.4.2 CNN-LSTM

Here, we briefly introduce the components of a CNN-LSTM, and it's significance in our model. DLP-LES comprises Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) neural networks [28]. CNN has been proved to be successful in image-related tasks including image classification, object detection, and

computer vision. Here, we take the advantage of CNN model in extracting heuristic features of the subgraph's adjacency matrices, which can be treated as an image in DLP-LES. In other words, we can transform the input data into an image to use in CNN. The LSTM model has been proved to be extremely effective in capturing long-term temporal correlations with arbitrary length. It can be used in several other applications, including text classification, handwriting recognition and speech recognition. The LSTM model preserves long-term dependencies effectively using three different gates:input gate activation (i_t), output gate activation (o_t) and forget gate (f_t). Its unit has a memory (c_t) cell, and its neuron input and output are x_t and h_t respectively at time step t.

$$LSTM = \begin{cases} i_t : \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \\ f_t : \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \\ o_t : \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_{t-1} + b_o) \\ g_t : \tanh(W_{xg}x_t + W_{hg}h_{t-1} + b_g) \\ c_t : f_t \odot c_{t-1} + i_t \odot g_t \\ h_t : o_t \odot \tanh(c_t) \end{cases} \tag{8.7}$$

where σ specifies the sigmoid activation function, bs denotes the bias, and Ws specify weight.

8.5 Architecture of DLP-LES Model

In this section, we describe our DLP-LES framework for link prediction. The proposed model has the following four major steps:

1. Link features lookup table construction.

2. Subgraph extraction and labeling.

3. Features matrix construction for links in subgraphs.

4. Modeling with CNN and LSTM.

At the beginning, we have a dynamic network $\mathcal{G}$ with the information of source and destination nodes that are observed within a time stamp T. Before we use this network, it needs to be arranged as a series of snapshots with equal time intervals $\Delta(t)$. In our case, $\{\mathcal{G}t_k, \mathcal{G}_{t-k+1}, \ldots, \mathcal{G}_{t-1}\}$ is treated as a sample first k snapshots with equal intervals as the input and the last $(t)^{th}$ snapshot as the output.

We name our proposed framework DLP-LES (**D**ynamic network **L**ink **P**rediction by **L**earning **E**ffective **S**ubgraphs), to highlight our focus on efficient common neighbor based subgraph to handle dynamic link prediction.

8.5.1 Link Features Lookup Table Construction

In this framework, features of links play a significant role to predict links in the future. As described in the above section, we have the sequence of snapshots of a given dynamic network $\mathcal{G}$. For the last snapshot $\mathcal{G}_t$, we extract subgraphs of the targeted links for prediction and analyze the heuristic features of each link in the subgraph. Evaluating heuristic features of links might be a repetitive process if we consider two targeted links from the subgraphs which have common links.

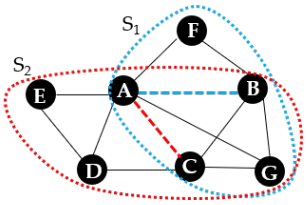


Figure 8.2: The representation of two subgraphs with common links for different targeted links. Subgraph S_1 with blue line is for the target link AB while Subgraph S_2 with red line is for the target link AC.

For example, consider two subgraphs S_1 and S_2 for the targeted links AB and AC as shown in figure 8.2. In our case, we need to calculate heuristic features for S_1 of links

$\{AB, AF, FB, AC, AG, CG, BC, BG\}$ and S_2 of links $\{AB, AE, AD, AC, AG, ED, DC, CG, BG, BC\}$. We need to repeat the calculation of feature for the common links $\{AB, AC, AG, CG, BC, BG\}$.

To avoid this repeated process, we initially build a lookup table $\langle \mathcal{R} \rangle$ to store the following information for every links of a given network G.

$$\langle v, u \rangle = \begin{cases} \text{Minimum Number of Hops} \\ \text{Average Path Weight} \\ \text{Time Stamp:} \begin{cases} [t_k \langle cn, jc, aa, pa, ra, w \rangle] \\ [t_{k-1} \langle cn, jc, aa, pa, ra, w \rangle] \\ . \\ . \\ [t_t \langle cn, jc, aa, pa, ra, w \rangle] \\ [t_{t+1} \langle cn, jc, aa, pa, ra, w \rangle] \end{cases} \end{cases}$$

where $t_k \langle cn, jc, aa, pa, ra, w \rangle$, specifies the heuristic feature values of snapshots at t_k, and $\langle k, (k-1), \ldots, t, (t+1) \rangle$ specifies the series of time. The minimum number of hops represents the number of minimum hops between v and u from the last snapshot $t+1$, and the average path weight is the ratio between path weight of minimum hop and number of minimum hops from the last snapshot $t+1$. The average path weight of a link can be calculated as below,

$$w_{avg} \langle v, u \rangle = \frac{1}{2} \left(\frac{1}{h} \sum_{p=0}^{h} w_p \right) \tag{8.8}$$

where w_p specifies the shortest path distance between v and p, add up to node u and h represents the number of hops between v and u.

Our primary objective is to construct a repository $\langle \mathcal{R} \rangle$ that can be used to retrieve

information of links without calculating it repeatedly. Rather than accessing the repository as a table, using a hashing function to access the information has more benefits. For this purpose, we formulate the following hashing function.

$$f(\langle v, u\rangle|\langle u, v\rangle) = \langle \mathcal{R}\rangle \tag{8.9}$$

where v and u are any vertices and the hashing function can provide the feature information for any order of vertex pair. We first convert the node pair $\langle v, u\rangle$ to a unique key, which is the same key for the nodes v and u in any order (ie $\langle v, u\rangle$, $\langle u, v\rangle$). To collect any information that we store in a lookup table, we can use the above function 8.9. So, the complexity is O(1) to gather information of a given link.

8.5.2 Subgraph Extraction and Node Labeling

Another primary process of our model is subgraph extraction. Although few subgraph extraction methods are proposed in the existing literature [17, 18], the extracted subgraphs using existing methods for LP do not have sufficient information. We use common neighbors of any targeted nodes v and u to create subgraphs. The common neighbors can be collected from different hops of both nodes, v and u. Rather than collecting just neighbor nodes, collecting common neighbors of both nodes v and u will have more information to decide the existence of link between them in the future. We set a threshold value Θ to keep the number of nodes limit in the subgraph.

Definition 8.2. ***(Subgraph based on Common neighbors)*** *For a dynamic network of last snapshot $\mathcal{G}_t = \langle \mathcal{V}, \mathcal{E}_t\rangle$, $\mathcal{G}' = \langle \mathcal{V}', \mathcal{E}'_t\rangle$ is a subset of sets of common neighbor nodes of two nodes $v_i \in \mathcal{V}'$ and $v_j \in \mathcal{V}'$, and are denoted as $\Gamma(v_i)$ and $\Gamma(v_j)$ if and only if $\mathcal{V}' \subseteq \mathcal{V}$ and $\mathcal{E}'_t \subseteq \mathcal{E}_t$, $\mathcal{V}'$ is a set of common neighbors for the targeted links, and $|\mathcal{V}'| = \Theta$.*

The algorithm 4 shows the step by step procedure to extract subgraph $\mathcal{G}'$ for a given link between v and u from a dynamic network of last snapshot $\mathcal{G}_t$. Since dynamic

networks evolve over time, most recent snapshot has more reliable information for the link predictions in the future [29]. At the beginning, the first order common neighbors $\Gamma^1(i) \cap \Gamma^1(j)$ of v and u are collected and stored to a node list N_Θ. Then, gradually increase the order of common neighbors $(\Gamma^2(i) \cap \Gamma^2(j)), (\Gamma^3(i) \cap \Gamma^3(j)), ...,$ until $|N_\Theta| \geq \Theta$, where $\Gamma^p(q)$ is the p^{th} order neighbor nodes of node q.

Algorithm 4 Common Neighbor Based Subgraph Extraction

Input: Target link E_{vu}, a snapshot graph $\mathcal{G}_p = \langle(\mathcal{V}, \mathcal{E}_p)$ at time p.
Output: Subgraph $\langle G' \rangle$ for the link E_{vu}.

$N_\Theta = \{v, u\}$
$N_{temp} = \{\}$
$h = 1 \leftarrow$ number of order
while $|N_\Theta| \leq \Theta$ **do**
    $N_{temp} = \Gamma^h(v) \cap \Gamma^h(u)$
    $N_\Theta = N_\Theta \cup N_{temp}$
    $h \leftarrow h + 1$
end while
$\langle G' \rangle \leftarrow$ subgraph $G(N_\Theta)$
return $\langle G' \rangle$

The above procedure may return the number of node list of the subgraph more than the defined threshold limit, $|N_\Theta| > \Theta$. At this point, each extracted subgraphs of last snapshot of a given dynamic network may have different number of node list. This inconsistency situation creates problem when we train convolutional neural network. We solve this issue by removing some nodes when we process labeling to keep the number of nodes limits equivalent to Θ.

Node labeling is another significant process in this study. It helps to maintain the consistency of the subgraphs. After we extract the subgraph, the nodes containing the target link get the labels 1 and 2. We use the node list N_Θ, which returns from subgraph extraction. We then remove the nodes belonging to the targeted link from N_Θ. To order the remaining nodes $R_\Theta = N_\Theta - \{1, 2\}$, we use the information of average minimum hops and average path weight. We can use the hashing function

Algorithm 5 Subgraph Node Labeling

Input: Nodes List N_Θ, Target link E_{vu}, Subgraph $\langle \mathcal{G}' \rangle$
Output: Ordered nodes list O_Θ

$O_\Theta = \{v, u\}$
$R_\Theta = N_\Theta - \{v, u\}$
$M \leftarrow$ Map for node information
for all $i \in R_\Theta$ **do**
  $h_{v,i}, w_{avg}\langle v, i\rangle = f\langle v, i\rangle$
  $h_{u,i}, w_{avg}\langle u, i\rangle = f\langle u, i\rangle$
  $w^i_{avg} = \frac{1}{2}(w_{avg}\langle v, i\rangle + w_{avg}\langle u, i\rangle)$
  $h^i_{avg} = \frac{1}{2}(h_{v,i} + h_{u,i})$
  $M \leftarrow (encode(i, 1/w^i_{avg}, h^i_{avg}))$
end for
sort M
for i in M **do**
  $O_\Theta \leftarrow O_\Theta \cup i$
  if $|O_\Theta| = \Theta$ **then**
    break
  end if
end for
return $\langle O_\Theta \rangle$

equation 8.9 to get the required information. However, we need an average number of hops (H_{Avg}) and average path weight (W_{Avg}). So, we use the following formulas to evaluate H_{Avg} and W_{Avg}.

$$H_{avg}\langle v, u\rangle = \frac{1}{2}(h_{v,i} + h_{i,u}) \tag{8.10}$$

$$W_{avg}\langle v, u\rangle = \frac{1}{2}(w_{avg}\langle v, i\rangle + w_{avg}\langle i, u\rangle) \tag{8.11}$$

where $h_{v,i}$ (resp. $h_{i,u}$) is the minimum number of hops between v and i (resp. $h_{i,u}$), and $w_{avg}\langle v, i\rangle$ (resp. $w_{avg}\langle i, u\rangle$) is the average path weight of the link $\langle vi\rangle$ (resp. $\langle iu\rangle$).

Our aim is to order the nodes in R_Θ in a consistent way. We can sort them first with average hop in ascending order and then with average path weight in descending order which helps to break tie from first ordering. Therefore, we come up with an idea to encode both H_{avg} and W_{avg} into single form which reduces the complexity.

For example, to generate a encoder to the node with $H_{avg} = 1.5$ and $W_{avg} = 1.75$, we encode as shown in figure 8.3, where the first portion indicates the value of $H_{avg} = 1.5$ and last portion indicates the reciprocal value of $W_{avg} = 1.75$, which remains always within the range $0 \ < 1/W_{avg} \leq 1$ in our case.

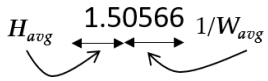


Figure 8.3: Encoding Format Example: first portion of the encoder specifies the average hop (H_{avg}) and the last portion specifies the reciprocal of average path weight (W_{Avg}).

We now order remaining nodes list based on encoded value and store them until the total nodes equal to threshold value Θ. Algorithm 5 shows the step by step labeling process for labeling. For example, Figure 8.4 represents the process of subgraph node labeling. The leftmost figure illustrates a subgraph from the last snapshot of a given dynamic network for the target link $\langle 12 \rangle$. The nodes display the information of average hop (H_{Avg}) and average weight (W_{Avg}). We encoded these values to generate a unique code as shown in second rightmost Figure 5. Finally, every node gets a unique label after ordering encoding values.

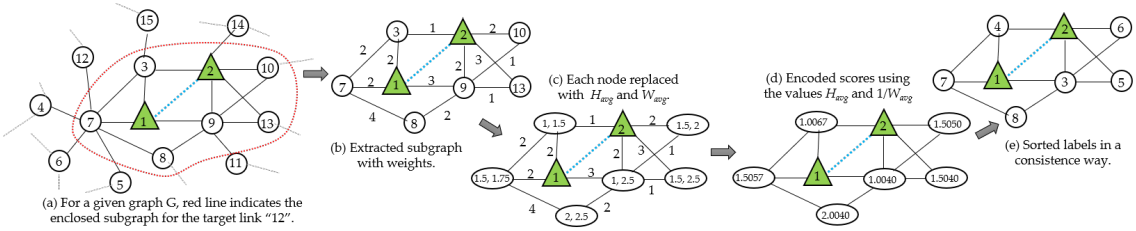


Figure 8.4: The representation of subgraph labeling based on encoding method. The left most figure represents the extracted subgraph with edge weight. Nodes with different colors indicates the various average hop distance (equal H_{avg} has same color), followed by average weight. The middle figure shows the encoded values and the right most figure represents the final labeling.

8.5.3 Feature Matrix Construction

We have a series of snapshots of a given dynamic network. The subgraph extraction and node labeling have been processed at the last snapshot t. The same subgraph should have evolved throughout the time series $t_k, t_{k-1}, \ldots, t_t$. We therefore construct feature metrics for a subgraph of each snapshot. As we already discussed in the previous section, we build feature matrices of CN, JC, AA, PA, RA and Weight.

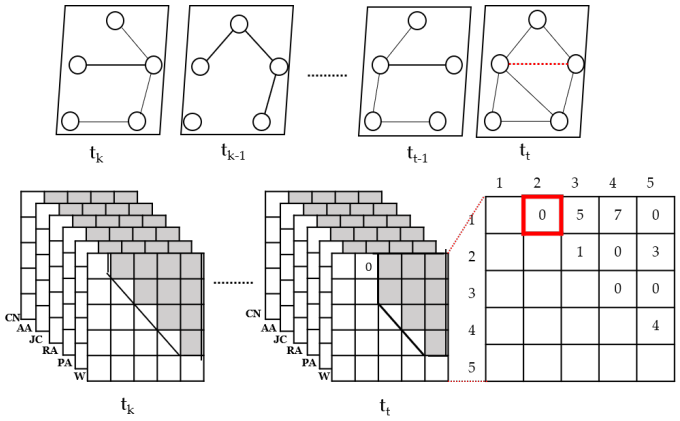


Figure 8.5: Example: (top) the representation of how a subgraph evolving through each snapshot, and (bottom) the way how adjacency matrices are created and the enlarged form of adjacency matrix for weight graph.

In figure 8.5, top figures illustrates the way how a subgraph evolves through time. We keep the same vertices of the subgraph in every snapshot and examine the evolution of links. We need to construct 6 feature matrices for the subgraph in each snapshot. Totally we construct $6 \times k$ number of feature matrices for a subgraph, where k is the time steps considered in our case.

Algorithm 6 describes the process of feature matrices construction. We create empty adjacency matrix lists to store the features of CN, JC, AA, PA, RA, W in

each snapshot as below;

$$\{A^k_{l\times l}\langle cn\rangle, A^k_{l\times l}\langle jc\rangle, A^k_{l\times l}\langle aa\rangle, A^k_{l\times l}\langle pa\rangle, A^k_{l\times l}\langle ra\rangle, A^k_{l\times l}\langle w\rangle\}$$

$$\{A^{k-1}_{l\times l}\langle cn\rangle, A^{k-1}_{l\times l}\langle jc\rangle, A^{k-1}_{l\times l}\langle aa\rangle, A^{k-1}_{l\times l}\langle pa\rangle, \ldots, A^{k-1}_{l\times l}\langle w\rangle\}$$

$$\ldots$$

$$\{A^{t}_{l\times l}\langle cn\rangle, A^{t}_{l\times l}\langle jc\rangle, A^{t}_{l\times l}\langle aa\rangle, A^{t}_{l\times l}\langle pa\rangle, \ldots, A^{t}_{l\times l}\langle w\rangle\}$$

where l is the size of the ordered nodes list of the subgraph. The algorithm 6 continues until the above empty lists are filled by fetching the required information of node list from the lookup table.

In each last snapshot of the adjacency matrix of the weighted graph, we assign zero to the positive target link to hide the information of link existence. In Figure 8.5, the bottom figure illustrates how we construct the adjacency matrices. We fill only the upper triangle of the matrix to avoid duplicate values. At the last snapshot, we indicate with a red box where the value is always zero.

Algorithm 6 Feature Matrix Construction

Input: Nodes ordered List O_Θ, Lookup Table $\langle\mathcal{R}\rangle$
Output: Feature Matrices $\langle\mathcal{F}\rangle = F_k\langle cn, jc, aa, pa, ra, w\rangle$,
$F_{k-1}\langle cn, jc, aa, pa, ra, w\rangle, \ldots F_t\langle cn, jc, aa, pa, ra, w\rangle$

1: $l \leftarrow |O_\Theta|$
2: $A^k_{l\times l}[\,], A^{k-1}_{l\times l}[\,] \ldots A^t_{l\times l}[\,] = \{\}$
3: **for** $i \in O_\Theta$ **do**
4: **for** $j - i \in O_\Theta$ **do**
5: $\left.\begin{array}{l} A^k_{l\times l}[\,] = F_k\langle cn, jc, aa, pa, ra, w\rangle \\ A^{k-1}_{l\times l}[\,] = F_{k-1}\langle cn, jc, \ldots, w\rangle \\ \ldots \\ A^t_{l\times l}[\,] = F_t\langle cn, jc, aa, pa, ra, w\rangle \end{array}\right\} = F\langle i, j\rangle$
6: **end for**
7: **end for**$\langle\mathcal{F}\rangle$

8.5.4 Modeling with CNN and LSTM

DLP-LES uses CNN and LSTM to model the link prediction framework. As described in the previous section, we have a sequence of adjacency matrices for a subgraph of a targeted link for the prediction from the last snapshot of a given dynamic network. The input data of DLP-LES is a sequence of adjacency matrices which is constructed as a form of $\Theta \times \Theta \times h$, where Θ is the number of nodes of the subgraph and h is the number of heuristic features used in our model. In DLP-LES, we treat each adjacency matrix as an image. We have the sequence of adjacency matrices in $t_k, t_{k-1}, \ldots, t_t$ as shown in Figure 8.5 bottom one. A sequence of images are really a video. So we can treat our model as a video classification problem, where positive and negative links are two different classes. The positive links represent the link existence, $(v_i, v_j) \in \mathcal{E}_t$ while the negative links represent the absence of links between any two nodes, $(v_i, v_j) \notin \mathcal{E}_t$. To train the classifier, we build a dataset using last snapshot of the dynamic network with all existing links for the positive link class and the same number of non-existing links by using downsampling technique.

CNN is well known for image classification. We leverage this character to learn and extract features from each image, in our case each adjacency matrix. We first feed the input data to convolutional layers to extract the features and then pass those sequences to a separate LSTM to learn the long-range temporal dependencies from input sequences. In the CNN model, we use Rectified Linear Units (ReLu) as the activation function, which is computed using $f(x) = max(0, x)$, where, x is the input data. In the LSTM model and the output layer, we use sigmoid activation function, $\sigma(x) = \frac{1}{1+e^{(-x)}}$. In DLP-LES, we assign the binary cross-entropy for loss function to measure the performance of a classification model. It can be written as $-(y.log(p) + (1 - y).log(p))$, where y is the label, p is predicted probability.

8.6 Experimental Results

To evaluate the effectiveness of our model, we perform experiments with five real-world dynamic social networks.

8.6.1 Datasets

We use five benchmark real world dynamic networks to test our model: **Enron** corpus [30] is an email communication network from the senior management of Enron for 6 months with 151 nodes and 50571 edges. Each node represents an employee and link represents email sent among employees. **Radoslaw** [31] is also an email communication network of a mid-sized manufacturing company from 2010-01-01 to 2010-09-30 with 167 nodes and 82900 edges. **Contact** [32] represents data from wireless devices carried by people with 274 nodes and 28200 edges. Every node specifies people, and a link appeared when they contacted with a timestamp which recorded every 20 seconds for 4 days. **College Messages** [33] contain private messages sent on an online social network at the University of California, Irvine with 1899 nodes and 59835 edges. The edge has the timestamp t, the time any two people contacted each others. **EU-Core** [34] is an email information from a large European research institution with 986 nodes and 332334 links. The node represents the members from 4 different departments and the links are the communications among them.

8.6.2 Performance Evaluation

We evaluate the effectiveness of DLP-LES model by comparing it with simple heuristic methods: CN, JC, AA, PA and network embedding methods: DeepWalk, node2vec, LINE and SDNE.

1. DeepWalk [13]: Random walks is used to learn latent representations, and considers vertices from second order proximity.

2. node2vec [12]: It learned by mapping of nodes to a low-dimensional space of features to maximizes the probability of preserving network proximity of nodes.

3. LINE [14]: It is suitable for any type of networks, including large scale networks. It used edge-sampling method to learn both the local and global network structures.

4. SDNE [35]: It is a semi-supervised deep model, and used both the first-order and second-order proximities together in an autoencoder based deep model.

For the implementation of the network embedding methods, we use the original source code by the author for node2vec[1], LINE[2], DeepWalk[3] and SDNE[4].

Evaluation Metric: Area Under the Curve (AUC) is the standard evaluation metric in both static and dynamic link prediction problem. AUC estimates the probability that the predictor gives a higher score to a randomly chosen positive link than a randomly chosen negative link. The larger the AUC is, the better the model performs. The AUC can be defined as,

$$AUC = \frac{n' + 0.5n"}{n} \tag{8.12}$$

where n specifies the number of Independence comparisons, n' specifies the number of times that the positive link gets a higher probability score than the negative link, and n" specifies the number of times when they are equal.

8.6.3 Experimental Procedure

We use Python3 to implement the DLP-LES model. The data processing is conducted on IBM cluster with the specification of POWER8 52 processor 256 GB of RAM. We

[1]https://github.com/aditya-grover/node2vec
[2]https://github.com/tangjianpku/LINE
[3]https://github.com/phanein/deepwalk
[4]https://github.com/xiaohan2012/sdne-keras

trained and tested our model in an Nvidia GTX 1050Ti, 4 GB GPU with 768 CUDA cores.

In DLP-LES, the CNN model contains 32 filters of size 5×5 in the convolution layer. Afterwards, it is sent to the average pooling with the size 3×3. The output is flattened before feeding into the LSTM model. The LSTM layer uses 128 internal cells. We train our neural network for 100 epochs with Adam optimizer algorithm. We assign 80% as training set, 10% as validation set, and 10% as testing set.

Dataset	**CN**	**JC**	**AA**	**PA**	**node2vec**	**LINE**	**DeepWalk**	**SDNE**	**DLP-LES**
Enron	0.8106	0.8751	0.8970	0.8442	0.7596	0.5042	0.7190	0.9437	**0.9769**
Radoslaw	0.8417	0.8307	0.9028	0.8753	0.7417	0.6153	0.7342	0.8709	**0.9330**
Contact	0.8457	0.9141	0.9142	0.9027	0.8741	0.7360	0.8451	0.9376	**0.9913**
CollegeMessages	0.5742	0.5774	0.5843	0.5901	0.7049	0.4905	0.7506	0.7806	**0.9852**
EU-core	0.9227	0.9302	0.9341	0.7553	0.8602	0.6587	0.8201	0.9574	**0.9729**

Table 8.1: Comparison of AUC with standard baseline methods for dynamic network link prediction.

8.6.4 Results

The performance of the experimental setup for DLP-LES using CNN-LSTM is represented in Table 8.1. The results are measured based on AUC in various benchmark dynamic datasets. DLP-LES outperforms all the standard state-of-the-art methods and most common heuristic methods. It also achieves above 97% of AUC in all tested dynamic networks except Radoslaw. However, DLP-LES reaches 93% of AUC in Radoslaw, which is higher than other compared methods. In College message, DLP-LES is remarkably outperformed than all baseline methods while others reach up to 78%.

The computational complexity of DLP-LES model can be expressed based on several factors. As we already explained in the above section, we first evaluate the heuristic values of each links in the given networks and stored as a lookup table. The complexity of subgraph extraction process is $O(kn)$, where k is the average hop

number for the subgraph and n is the number of links. The complexity of feature matrix construction is $O(sn)$, because our feature matrix algorithm collects required information from the lookup table, which is $O(1)$, where $s = \Theta$ is the number of nodes in the subgraph.

8.7 Conclusions

In this paper, we propose a novel framework DLP-LES, which is for link prediction problem in dynamic social networks based on effective subgraphs. This model uses the heuristic features of common neighbor based subgraph and learns the evolving pattern throughout the considered time to predict future links. In this work, we introduce an encoding method for labeling subgraph consistently. To reduce the complexity, we construct a lookup table with all required information of links to use frequently through our proposed hash function. We also propose an algorithm for feature matrix construction, which is thereafter feed into CNN to extract the features and send to LSTM to learn long-term temporal feature of dynamic network. Since it analyzes the subgraph of a target link, this model has the advantage in applying for large-scale networks. We evaluate the performance of our model against the state-of-the-art methods and the basic heuristic method. DLP-LES achieves significantly high improvement than compared methods. Further, DLP-LES opens a new research direction in temporal network compression and expanding community detection in dynamic networks.

Acknowledgment

This research work was supported by International Business Machines (IBM); experiments were conducted on a high performance IBM Power System S822LC Linux Server.

Bibliography

[1] T. Y. Berger-Wolf and J. Saia, "A framework for analysis of dynamic social networks," in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2006, pp. 523–528.

[2] M. Kazemilari and M. A. Djauhari, "Correlation network analysis for multi-dimensional data in stocks market," *Physica A: Statistical Mechanics and its Applications*, vol. 429, pp. 62–75, 2015.

[3] L. Wang and J. Orchard, "Investigating the evolution of a neuroplasticity network for learning," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2017.

[4] L. A. Adamic and E. Adar, "Friends and neighbors on the web," *Social networks*, vol. 25, no. 3, pp. 211–230, 2003.

[5] H. Sid Ahmed, B. Mohamed Faouzi, and J. Caelen, "Detection and classification of the behavior of people in an intelligent building by camera." *International Journal on Smart Sensing & Intelligent Systems*, vol. 6, no. 4, 2013.

[6] X. Wang, N. Gulbahce, and H. Yu, "Network-based methods for human disease gene prediction," *Briefings in functional genomics*, vol. 10, no. 5, pp. 280–293, 2011.

[7] M. E. Newman, "Clustering and preferential attachment in growing networks," *Physical review E*, vol. 64, no. 2, p. 025102, 2001.

[8] L. Katz, "A new status index derived from sociometric analysis," *Psychometrika*, vol. 18, no. 1, pp. 39–43, 1953.

[9] X. Li, N. Du, H. Li, K. Li, J. Gao, and A. Zhang, "A deep learning approach to link prediction in dynamic networks," in *Proceedings of the 2014 SIAM International Conference on Data Mining*. SIAM, 2014, pp. 289–297.

[10] M. Rahman and M. Al Hasan, "Link prediction in dynamic networks using graphlet," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2016, pp. 394–409.

[11] J. Chen, J. Zhang, X. Xu, C. Fu, D. Zhang, Q. Zhang, and Q. Xuan, "E-lstm-d: A deep learning framework for dynamic network link prediction," *arXiv preprint arXiv:1902.08329*, 2019.

[12] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2016, pp. 855–864.

[13] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2014, pp. 701–710.

[14] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "Line: Large-scale information network embedding," in *Proceedings of the 24th international conference on world wide web*. International World Wide Web Conferences Steering Committee, 2015, pp. 1067–1077.

[15] M. Kaya, M. Jawed, E. Bütün, and R. Alhajj, "Unsupervised link prediction based on time frames in weighted–directed citation networks," in *Trends in Social Network Analysis*. Springer, 2017, pp. 189–205.

[16] C. Chiu and J. Zhan, "Deep learning for link prediction in dynamic networks using weak estimators," *IEEE Access*, vol. 6, pp. 35 937–35 945, 2018.

[17] M. Zhang and Y. Chen, "Weisfeiler-lehman neural machine for link prediction," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2017, pp. 575–583.

[18] Zhang, Muhan, and Y. Chen, "Link prediction based on graph neural networks," in *Advances in Neural Information Processing Systems*, 2018, pp. 5165–5175.

[19] K.Ragunathan, K.Selvarajah, and Z.Kobti, "Link prediction by analyzing common neighbors based subgraphs using convolutional neural network." in *ECAI*, 2020.

[20] Yao, Lin, L. Wang, L. Pan, and K. Yao, "Link prediction based on common-neighbors for dynamic social network," *Procedia Computer Science*, vol. 83, pp. 82–89, 2016.

[21] S. Gao, L. Denoyer, and P. Gallinari, "Temporal link prediction by integrating content and structure information," in *Proceedings of the 20th ACM international conference on Information and knowledge management*. ACM, 2011, pp. 1169–1174.

[22] W. Yu, W. Cheng, C. C. Aggarwal, H. Chen, and W. Wang, "Link prediction with spatial and temporal consistency in dynamic networks." in *IJCAI*, 2017, pp. 3343–3349.

[23] X. Ma, P. Sun, and Y. Wang, "Graph regularized nonnegative matrix factorization for temporal link prediction in dynamic networks," *Physica A: Statistical mechanics and its applications*, vol. 496, pp. 121–136, 2018.

[24] S. De Winter, T. Decuypere, S. Mitrović, B. Baesens, and J. De Weerdt, "Combining temporal aspects of dynamic networks with node2vec for a more efficient dynamic link prediction," in *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. IEEE, 2018, pp. 1234–1241.

[25] T. Li, J. Zhang, S. Y. Philip, Y. Zhang, and Y. Yan, "Deep dynamic network embedding for link prediction," *IEEE Access*, vol. 6, pp. 29 219–29 230, 2018.

[26] K. Lei, M. Qin, B. Bai, G. Zhang, and M. Yang, "Gcn-gan: A non-linear temporal link prediction model for weighted dynamic networks," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 2019, pp. 388–396.

[27] C. A. Bliss, M. R. Frank, C. M. Danforth, and P. S. Dodds, "An evolutionary algorithm approach to link prediction in dynamic social networks," *Journal of Computational Science*, vol. 5, no. 5, pp. 750–764, 2014.

[28] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[29] J. Chen, X. Xu, Y. Wu, and H. Zheng, "Gc-lstm: Graph convolution embedded lstm for dynamic link prediction," *arXiv preprint arXiv:1812.04206*, 2018.

[30] B. Klimt and Y. Yang, "The enron corpus: A new dataset for email classification research," in *European Conference on Machine Learning*. Springer, 2004, pp. 217–226.

[31] R. Rossi and N. Ahmed, "The network data repository with interactive graph analytics and visualization," in *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[32] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott, "Impact of human mobility on opportunistic forwarding algorithms," *IEEE Transactions on Mobile Computing*, no. 6, pp. 606–620, 2007.

[33] P. Panzarasa, T. Opsahl, and K. M. Carley, "Patterns and dynamics of users' behavior and interaction: Network analysis of an online community," *Journal of the American Society for Information Science and Technology*, vol. 60, no. 5, pp. 911–932, 2009.

[34] A. Paranjape, A. R. Benson, and J. Leskovec, "Motifs in temporal networks," in *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*. ACM, 2017, pp. 601–610.

[35] D. Wang, P. Cui, and W. Zhu, "Structural deep network embedding," in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, 2016, pp. 1225–1234.

Chapter 9

Summary, Conclusions and Future Directions

Motivated by the observations from the existing researches in team formation problems in social science and management and critical challenges in computational models, in this dissertation, we provide a generic computational framework for team formation problems in social networks. This research investigates the team formation problem from two different perspectives: discovering teams of experts to recommend for a set of tasks (in the first six chapters) and predicting new members who can join teams in the future (in the last two chapters).

We consider three different networks to examine team formation problems: co-authorship networks (chapter 2), healthcare setting (chapters 3 and 4), and industry organizational setting (chapter 5). We begin our research on expert networks that contain professionals who have skills and expertise in particular areas. On the other hand, enterprises and academic staffs search to find talent and expertise from such networks to complete projects or tasks. We aim to search the network to identify an optimal set of experts covering the required skills while keeping the communication cost to a minimum, which measures the performance of teams. We examine

communication costs using the shortest path distance and diameter distance in chapter 1. Our major contribution in this work is to attempt knowledge-based cultural algorithms in the team formation problems to reduce the computational challenges. The knowledge-based cultural algorithm is an evolutionary searching technique, which uses higher-order cultural evolution to reduce the search domain by extracting knowledge and updating it in each generation. The individual representation, knowledge extraction, and utilizing them in each evolution are the main contributions of this research work. Our method shows the highest accuracy when compared to the other algorithms and is the closest to the exact algorithm (exhaustive search).

In chapters 3 and 4, we attempt the healthcare settings, especially to assemble a team of care providers for patients in community-oriented palliative care. The main objective of this research is to optimize the patient's care services and human resource allocation process. In palliative care, we have a group of patients with needs who are not able to perform some of their ordinary life activities due to their limited capabilities, as a consequence of their disease or disorders. On the other hand, we have a group of care providers who are capable, skilled, and ready to provide a wide range of services to the patients to fulfill those needs. We propose a framework to tackles the challenges of assigning members to a team of care providers in an optimal manner to help the patient satisfy their needs while taking into consideration the communication, distance, and contact costs. Chapter 4 provides the visualization of the problem discussed in chapter 3. Our model provides the most effective solutions compared to other methods.

In chapter 5, we tackle the problem of finding the profit-maximizing cluster hires. This research is a little different from the previous chapters. Organizations or online freelancers search for the most cost-effective teams to fulfill their goals. The recruiting experts expect a salary to perform a set of tasks. The objective of this research is to identify teams to maximize the profit of tasks under a given budget using the

knowledge-based cultural algorithm while optimizing communication cost, the profit of projects, productivity, and load limit. We implement project greedy, expert greedy, genetic algorithms, and exact algorithms and compare our results. Our approach outperforms other methods and reaches near to the exact method.

Chapter 6 provides a unified framework for the team formation problem in dynamic social networks. We propose two temporal based cost functions: dynamic communication cost and dynamic expertise level. With these cost functions, we engage the time difference from current to considered time. Our proposed cost functions show a high impact on team formation problems. In addition to this, we incorporate some essential ideas from management perspectives. We examine the trust score based on emotional intelligence index, profile similarity, and explicit score. This model is a major contribution to the research in team formation problems since it formulates the team formation problem in dynamic environments and models as a multi-objective optimization problem to optimize dynamic communication cost, dynamic expertise level, geological proximity, and collective trust score. We apply the multi-objective cultural algorithms to find the non-dominated solution while extracting situational and topological knowledge from each generation to reduce the searching domain. Experimental results are compared against well-known non-dominated algorithm NSGA II and exact algorithms.

Chapters 7 and 8 are designed to predict future collaboration with existing teams of experts, which is technically an application of link prediction. Link prediction problem aims to examine whether a link between any two nodes appear in the future or not. This prediction mechanism is a classification problem that classifies existing links and non-existing links. In chapter 7, we examine the temporal behaviors of vertices to find active people in dynamic networks. We introduce a probability function to predict links that occur in the future to connect with other members based on the activeness of the person. We introduce a time-varying score function to evaluate the activeness

of vertices that uses the number of new interactions and the number of frequent interactions with existing connections. We also consider two additional objective functions in our model: a weighted shortest distance between any two nodes and a weighted common neighbor index. To handle this classification problem, we employ Multi-Layer perceptron, a deep learning framework. We train (80% of the dataset), validate (10%) and test (10%) our model, and compare our results in AUC (Area Under Curve) with well-known baseline machine learning approaches and heuristic methods. Our results reach above 93% of AUC.

In chapter 8, we use the existing heuristic methods to predict links in the future. Motivated by observing the challenges in existing methods, we introduce a novel framework to address the challenges in reaching high accuracy in various types of networks and handling computational costs. We use common neighbors based subgraph of a target link and learn the transitional pattern of it for a given dynamic network. We then extract a set of heuristic features of the evolving subgraph to gather additional information about the target link. To reduce computational costs, we introduce some mechanism: construct a lookup table with required information of links in the network, uses a hashing method to store and fetch link information, and introduce encoding method to label the nodes in subgraphs. Since we handle a sequence of snapshots of a dynamic network, the features of the extracted subgraph evolve in every snapshot. So we transform our problem as a video classification problem and apply Convolutional Neural Networks (CNN) to extract higher-level features of subgraph efficiently and Long Short-Term Memory (LSTM) neural networks to learn long-range dependencies of sequential data and capture the evolving patterns of the subgraph in the dynamic network. To verify the effectiveness of our model, extensive experiments are carried out on five real-world dynamic networks and compared those results against four network embedding methods and basic heuristic methods.

To conclude, we describe the complex and evolving nature of dynamic social net-

works to search and recommend teams of experts to produce successful outcomes of tasks under some constraints. To handle the above research problems, we use an integrated knowledge-based computational model, cultural algorithms in this dissertation. Our model outperforms the existing other approaches. Later, we aim to predict future collaborations. Since this problem is a classification problem, we attempt with deep learning frameworks in this dissertation. Our model performs well in prediction and achieves high accuracy.

This dissertation has several new paths to further research with various applications. For instance, in online video games, we can research how efficiently assign team members, and in a warehouse of distribution centers, efficiently assigning a group of robots is a challenging research direction. Moreover, our prediction framework can be expanded to handle community detection in dynamic networks. Another research directions are to focus on dynamic knowledge graph completions and dynamic recommender system.

Vita Auctoris

NAME:	Kalyani Selvarajah
PLACE OF BIRTH:	Point Pedro, Sri Lanka
EDUCATION:	Bachelor of Science
	Computer Science, Mathematics and Physics,
	University of Peradeniya, Sri Lanka.
	Master of Science in Computer Science,
	University of Peradeniya, Sri Lanka.
	Doctor of Philosophy in Computer Science,
	University of Windsor, Canada.