Cluster Hire in Social Networks Using Modified Weighted Structural Clustering Algorithm for Networks (MWSCAN)

Harshil Patal
University of Windsor

CLUSTER HIRE IN SOCIAL NETWORKS USING MODIFIED WEIGHTED STRUCTURAL CLUSTERING ALGORITHM FOR NETWORKS (MWSCAN)

By

Harshil Patel

A Thesis
Submitted to the Faculty of Graduate Studies
through the School of Computer Science
in Partial Fulfillment of the Requirements for
the Degree of Master of Science
at the University of Windsor

Windsor, Ontario, Canada

2021

CLUSTER HIRE IN SOCIAL NETWORKS USING MWSCAN: MODIFIED WEIGHTED STRUCTURAL CLUSTERING ALGORITHM FOR NETWORKS

by

Harshil Patel

APPROVED BY:

H. Wu
Department of Electrical and Computer Engineering

S. Samet
School of Computer Science

D. Wu, Advisor

School of Computer Science

August 24, 2021

DECLARATION OF ORIGINALITY

I hereby certify that I am the sole author of this thesis and that no part of this thesis has been published or submitted for publication.

I confirm that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged as per 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 written permission from the copyright owner(s) to include such material(s) in my thesis and have added copies of such copyright clearances to my appendix.

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

ABSTRACT

The concept of effective collaboration within a group is immensely used in organizations as a viable means for improving team performance. Any organization or prominent institute, who works with multiple projects needs to hire a group of experts who can complete a set of projects. When hiring a group of experts, numerous considerations must be taken into account. In the Cluster Hire problem, we are given a set of experts, each having a set of skills. Also, we are given a set of projects, each requiring a set of skills. Upon completion of each project, a profit is generated for an organization. Each expert demands a monetary cost (i.e., salary) to provide his/her expertise in projects. The Cluster Hire problem can be solved by hiring a group of experts for a set of projects within the constraints of a budget for hiring and a working capacity of each expert. An extension to this problem is assuming there exists a social network amongst the experts, which contains their past collaboration information. If two experts have collaborated in the past, then they are preferred to be on the same team in the future. The goal of our research is to find a collaborative group of experts who can work effectively together to complete a set of projects. Currently, the solution to the Cluster Hire problem in social networks is achieved using greedy heuristic algorithms and Integer Linear Programming (ILP) approach. Greedy algorithms often generate fast results, but they make locally optimal choices at each step and do not produce global optimal results. The drawbacks of the ILP approach are that it requires a considerable amount of memory for the creation of variables and constraints and also has a very high processing time for large networks. Whereas, Weighted Structural Clustering Algorithm for Networks (WSCAN) has been proved to produce faster results for Team Formation Problem (i.e., hiring a team of experts for a single project), which is a special case of Cluster Hire problem. We are proposing to solve the Cluster Hire problem in social networks using Modified Weighted Structural Clustering Algorithm for Networks (MWSCAN). We run our experiments on a large dataset of 50K experts. ILP is not capable of working with such large networks. Therefore, we will be comparing our results with the greedy heuristic solution. Our findings indicate that the MWSCAN algorithm generates more efficient results in terms of the number of projects completed and profit produced for the given budget compared to the greedy heuristic algorithm to solve the Cluster Hire problem in social networks.

DEDICATION

Dedicated to my parents Mukeshbhai Patel and Pannaben Patel, my sister Pooja Patel and my adorable nephew Hetarth, and niece Aarvi

And also, the rest of my family and friends

ACKNOWLEDGMENTS

I owe a debt of gratitude to Dr. Dan Wu for the vision and foresight, which inspired me to conceive the thesis work. As my teacher and mentor, he has taught me more than I could ever give him credit for here. I am thankful to my thesis committee members, Dr. Saeed Samet and Dr. Huapeng Wu, for providing me extensive personal and professional guidance, which helped me learn a great deal about both scientific research and life in general.

Nobody has been more important to me in the pursuit of this thesis than the members of my family & friends. I would like to thank my parents; whose love and guidance are with me in whatever I pursue. They are the ultimate role models and the source of my inspiration. Most importantly, I wish to thank all the faculties and staff of the School of Computer Science and my friends. They provided continual support and encouragement through my course work at the University of Windsor.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

1.1 Overview

The vital aspect of any company/organization is the recruitment of employees who can work together efficiently. Companies that undertake multiple projects at the same time have to hire more and more employees. In the era of social networking, recruiters usually rely upon online social networks available, such as LinkedIn, Kaggle, GitHub, and DBLP, for the recruitment of employees. When hiring, recruiters must ensure that a formed team of experts will be cost-effective and efficient enough to finish the given projects in a time-sensitive manner.

A project is where a group of experts, each having required skills, work together in a team towards a common goal. A project contains a collection of tasks or activities, which are fulfilled by a group of experts. Additionally, an expert is somebody who has experience obtained through training and instruction in one or more skills. Undertaking a collection of tasks requiring a set of diversified skills needs a group of experts. This is achieved by allotting skills to various experts with complementary expertise. This requires big organizations to effectively hire a group of experts with a varied skill set to perform a collection of tasks to finish *a single project*, which is called a team formation problem (TFP) [6].

Whereas Cluster Hire aims to hire a group of experts that can complete all skills required for *a set of projects* instead of a single project. Each expert requires a monetary cost (i.e., salary) to participate in projects, so there must be a set budget to hire experts. The working capacity of an expert is another critical parameter while hiring an expert for multiple projects. Each expert is associated with the maximum working capacity they can offer, which means that an expert can only work on a certain number of projects at a time. The goal is to employ substantial experts who are efficient enough to complete a set of projects within the given budget and working capacity, which proves beneficial for the organization and institute. This problem is called Cluster Hire.

Golshan et al. [12] first introduced the Cluster Hire problem, where they discussed the problem in the context of the online labor market websites like Freelancer.com and Guru.com. For instance, large organizations usually work on numerous projects. A vast and variable skill set is required to accomplish these projects to generate profit. Therefore, they hire a group of experts who can finish these projects.

While finding a group of experts for a set of projects, we try to find experts who have collaborated in the past. So, they can work well together, and the hired group of experts is collaborative. Social networking can ensure that the formed group of experts is collaborative. Social networks such as DBLP and LinkedIn can provide us with information about the collaborating experts, which helps hire a collaborative group of experts [1]. In our research, we focus on the Cluster Hire problem in social networks. In our experiments, the social network is modeled as a graph where the experts are interlinked. Each node of the graph indicates the expert associated. An edge connects two experts, and edge weight represents their communication cost obtained from their past collaboration. The more past collaboration between experts, the less is the communication cost. Thus, the past collaboration between experts is directly linked to the communication cost of a formed group of experts. For instance, if two experts have worked together on eight projects in the past, and the second pair of experts have worked together on eleven projects in the past. Then, with the experience of eleven projects together, the pair will have less communication cost between them than the couple with eight projects. If two experts do not have any previous collaboration, communication cost is measured by the sum of the weight representing the shortest path between them. The hiring officer will also hire the experts with past collaboration because it resembles their proper working frequency and compatibility. If a group of experts can work well together, then it helps in completing projects on time. The company generates profit by completing projects. Therefore, we choose a group of experts with minimum communication cost to complete projects.

Keeping in mind the communication cost between two experts, expert's capacity, and budget assigned, we find a group of experts with the least communication cost for the given set of projects. All features of the Cluster Hire problem in social networks are described in detail in the latter part of the thesis.

1.2 Motivation

In the emerging technological world, many companies are growing worldwide, which works on the phenomena of collective expertise. Collective expertise can be defined as the occurrence of a certain level of knowledge in a group of people or experts who possess the skills required to complete projects. Companies benefit from various projects or opportunities available in the market that the company targets. For each project, they need to hire or allocate a group of experts. In this regard, effective resource allocation is critical for companies to meet specific objectives, like completing projects within deadlines and generating revenue. Cluster Hire in social networks can help fulfill these objectives by hiring experts effectively. It considers past collaboration between experts while hiring so the final group of experts is collaborative and can work effectively with each other to meet the deadlines [12].

Moreover, it is well said that time is money; therefore, Cluster Hire in social networks can help us hire a group of experts efficiently and faster than the conventional hiring methods. Social networks have linked the current day world and have made logistic advances easy. It can save a lot of time and resources companies spend on hiring employees. So, Cluster Hire in social networks can help companies fulfill their goals and make hiring experts easier.

Furthermore, online freelance businesses such as Freelancer, Fiverr, Upwork, and Guru usually work on multiple projects. These businesses hire freelancers with various skill sets, and freelancers work remotely on different projects. In other words, these projects have the required skills, and they have to hire experts or freelancers to complete these projects. For each skill of a project, employers employ an expert who can fulfill the skill requirement. There is also a constraint of the budget, which is allocated for hiring experts. So, the total sum of the salary of experts should be less than the defined budget.

Finally, there is much research done with the major objective of solving the Cluster Hire problem in social networks using greedy algorithms and Integer Linear Programming (ILP) [23, 31]. There are limitations to both approaches. Greedy algorithms often generate fast results, but they make locally optimal choices at each step and do not produce near-optimal or optimal results. Integer Linear Programming (ILP) is not ideal for large social networks due to its computational complexity because it requires considerable memory to create variables and constraints and has a very high processing time for large networks [31]. The later part of the section explains the problem and solution outline, scope, and structure of the thesis.

1.3 Problem and Solution Outline

Cluster Hire can be defined as forming a group of experts, each having the required skills for a set of projects under the given budget and working capacity. A variation to that is Cluster Hire in social networks, where along with hiring a group of experts having required skills for a given set of projects, we also make sure that selected experts have minimum communication cost with other experts in a group. If experts have collaborated in the past, they have less communication cost between them, which helps to complete projects effectively and on time. Now, we will discuss the terminologies to keep in mind while studying the Cluster Hire problem in social networks.

Firstly, each expert has their own *set of skills*. Secondly, experts have their *working capacity*. The *working capacity* indicates how many maximum projects can be assigned to that expert. This means that an expert can be allotted multiple projects at the same time based on their *skills* and *working capacity*. Thirdly, there is a *hiring cost* associated with each expert. The *hiring cost* of an expert is monetary value, which an expert expects to take part in projects. Fourthly, it requires a group of experts with specific skills to complete a *set of projects*, each project demanding particular skills. Lastly, there is a *social network* among the experts, which is retrieved from their past collaborations and work experiences. A *social network* is represented as a graph, where each node is an expert, and the edge connecting two experts has an edge weight representing *communication cost* obtained from experts' past collaboration history.

Also, a predetermined *budget* is set to hire a group of experts for a given set of projects. The sum of *hiring cost* of each expert in a group of experts has to be within the given *budget*. The solution to Cluster Hire in social networks is hiring a group of experts for a particular set of projects within a given budget and working capacity while minimizing the communication cost between the hired group of experts.

We propose to solve Cluster Hire in social networks using a Modified Weighted Structural Clustering Algorithm for Networks (MWSCAN). The problem statement and the approach are further explained in detail in Chapter 2.

1.4 The scope of the thesis

In the past, Kalyani et al. [21] proposed a Weighted Structural Clustering Algorithm for Networks (WSCAN) to solve the Team Formation Problem in social networks, which is a special case of Cluster Hire in social networks. In the Team Formation Problem, employers hire a team of experts covering all the skills required for a *single project*. In our research, we have a *set of projects* requiring a specific skill set for its completion. We have a set of experts with associated skills to fulfill the skill requirements for multiple projects. Each expert has a working capacity associated with it to limit the participation in a number of projects. We have the budget assigned to hire a group of experts for a given set of projects. Also, we have the social network graph, which connects the experts and stores the value of communication cost between experts.

The Cluster Hire in a social network was first studied by Meet et al.   23] to hire a group of experts for a specific set of projects with a specific skillset to generate maximal profit, where each project has associated profit, which is generated after completion of the project. Their study mentioned numerous attributes related to team members and a team. These attributes were as follows:

1. The workload capacity of an individual expert
2. The hiring cost/salary of the experts
3. The sum of hiring cost of the experts within the assigned budget
4. The profit of the projects
5. The past/previous collaboration between experts

They proposed two greedy algorithms and extracted their results by conducting experiments on a large graph of experts.

Our thesis proposes solving the Cluster Hire problem in social networks by using a Modified Weighted Structural Clustering Algorithm for Networks (MWSCAN). Our goal is to hire a group of experts for a set of projects with minimum communication cost within the given budget and working capacity. This is an extension of the proposed work in the past [21]. We apply the MWSCAN algorithm to address the Cluster Hire problem in a social network and compare our work with the greedy algorithms.

1.5 Structure of the thesis

The rest of the thesis is organized in the following manner.

In chapter 2, we discuss the basic concepts and related work in the field of Team Formation Problem, Cluster Hire in social networks, SCAN, WSCAN. Also, we review the approaches proposed in the past to solve the problem.

In Chapter 3, we give a detailed explanation of the problem statement. We describe our proposed approach, which includes a detailed description of the MWSCAN algorithm, its goals, and constraints, along with an example.

In Chapter 4, we describe the implementation details and the contribution of our proposed work, and the results showing the comparisons with the previous approaches.

In Chapter 5, we conclude the thesis work, explain the findings of the work along with setting up the field of opportunities for possible future work.

CHAPTER 2

BACKGROUND STUDY

2.1 Overview of the research

2.1.1 Team formation Problem (TFP)

Hiring a group of experts who have expertise in one or more skills that are needed for the completion of *a single project* is known as Team Formation. This aids the big industries and organizations to finish the projects on time successfully. It also helps the experts enhance and sharpen their skills and improve their position in the organization by excellent performance. The concept here is to gather individuals with expertise in the skills required to complete the project successfully. Different constraints and factors can be utilized to select individuals to work in a team for a project. For example, a similar background of the team members is one factor that benefits the team by making communication within the team more effective. This reduces conflicts and miscommunication amongst team members. Thus, there is a reduction in the amount of time required to produce an active working group since less is needed to adjust the working styles.

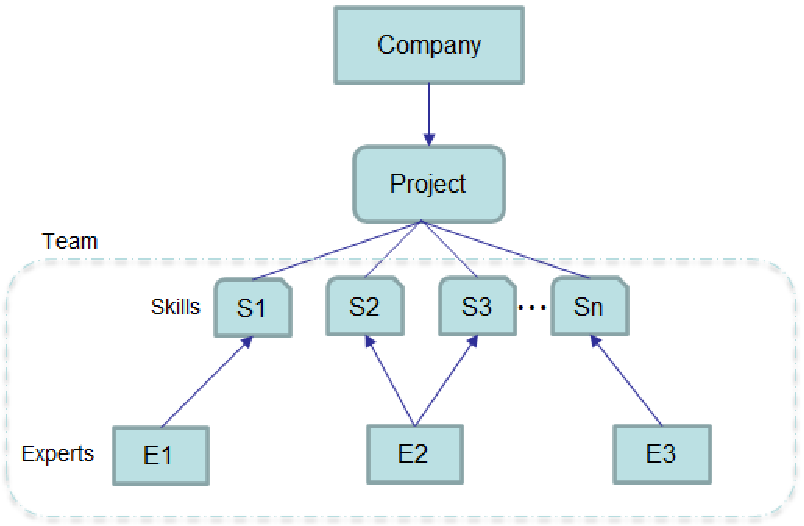


Figure 1: Team Formation

As described in Figure 1, there is one project, which requires a set of skills, and we hire a team of experts to fulfill all the skills needed to complete that project successfully.

2.1.2 Cluster Hire

Cluster Hire is the process of hiring a group of experts (individuals) to meet the requirements for *a set of projects* within the budget of the hiring organization and working capacity of experts. For instance, if a company needs to hire a group of experts for ten projects, where each expert possesses a predetermined skill set, the objective here would be to search for the experts with those skills required to successfully complete projects within the constraints. This is referred to as Cluster Hire problem.

There are various attributes and constraints that affect the selection process of the experts for a given set of projects in Cluster Hire problem, they are described below:

1) Budget

Budget is an upper limit monetary value assigned to a set of projects by the company's financial department. The goal is to hire group of experts within the budget for a set of projects. The sum of the hiring cost of experts for a set of projects cannot be more than the predetermined budget.

2) The experts and their attributes

The expert's skills: Each expert or individual in the social network working in a company is accredited for his/her expertise in a specific set of skills. In other terms, each expert has a set of skills he/she can offer to projects. Our goal is to select experts for specific projects depending on the skillset of the experts, which would help in completing projects and generating profit for the company.

The expert's hiring cost: The hiring cost can be defined as some monetary benefit associated with an expert for taking part in projects. Each expert has a predetermined hiring cost associated to it. The hiring cost remains unchanged irrespective of the number of projects assigned to experts or the number of skills experts have. It has to be taken into account that the sum of the hiring cost of the selected experts should not be higher than the budget assigned for hiring.

The expert's working capacity: Working capacity refers to the maximum number of projects an expert can be part of at a given time. In the scenario of an expert having many skills, which can fulfill the skill requirements of more than one project, we can assign multiple projects to expert within their working capacity. Please note that an expert can cover only one skill in a particular project; it cannot cover multiple skills of the same project. This is a reasonable assumption as organizations may need to hire more than one expert for the same skill in projects. For instance, if a project needs three experts in Java, then we need to hire three unique experts in Java for the same project instead of hiring one expert in Java with working capacity of three. But the project assignment for experts should be within their working capacity. For instance, if an expert has seven skills that are useful for the given set of projects, but the expert has a working capacity for four projects only. Then, the expert will be allotted four different projects for the four skills.

The experts' communication cost: In our setup, we have a social network, where experts are interconnected. The social network is represented as a graph, where nodes represent experts and nodes are connected via edges. The weight of the edges represents communication cost between experts. The communication cost between the experts depends on the number of instances when the experts have had past collaboration or work experiences. We ensure that the selected group of experts possess minimum communication cost, which is essential for good collaboration. A group of such experts reduces the overhead time spent in interpreting each expert's background etc.

3) The projects and their attributes

The Project's Skills: Each project requires predefined set of skills, which are essential to complete the project. The organizations hire individuals who possess the right level of expertise in the skills necessary to complete the project successfully. We must ensure that we hire no less than the number of experts required to fulfill the skill set needed for projects. For example, a project requires three experts for Machine Learning and two experts for Software Engineering. Our task is to find five different experts for the project for its successful completion.

The Project's Profit: Each project's profit is generated upon completion of the project by the hired group of experts. Each project has a monetary profit (predetermined revenue) associated with it, which the organization extracts after completing the project successfully.

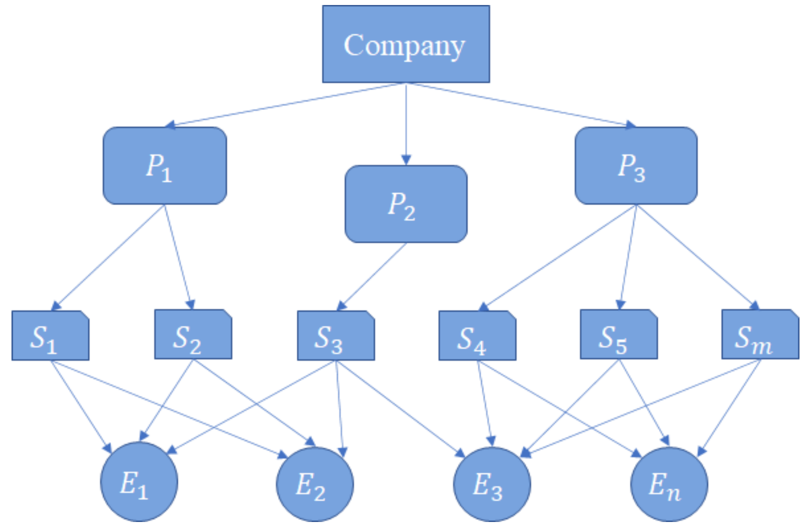


Figure 2: Cluster Hire

Table 1: Experts Information

Expert ID	**Skills of an expert**	**Hiring cost**	**Working capacity**
E_1	S_1, S_2, S_3	2000	3
E_2	S_1, S_2, S_3	4000	5
E_3	S_3, S_4, S_5, S_m	8000	4
E_n	S_4, S_5, S_m	5000	3

Figure 2 illustrates an example of the Cluster Hire problem. There is a company which works on three projects, namely project P_1, project P_2 and project P_3. Each project requires a specific set of skills for its completion. The goal is to hire a group of experts for the given three projects under the constraints of budget and working capacity. The budget assigned to hire a group of experts for these three projects is 20,000. If we take the project P_1, it requires one expert each for skill S_1 and skill S_2 as shown in Figure 2. So, based on the available experts shown in Table 1, we will hire experts E_1 and E_2 to complete project P_1. We can hire E_1 and E_2 as sum of their hiring cost is 6000, which is within the budget. Please note that we can hire the same expert for multiple projects keeping in mind the working capacity of an individual expert. Now, for project P_2, we just require one expert for skill S_3 but we have three experts, E_1, E_2 and E_3 who can fulfill that skill. Therefore, we can allocate one of the already hired expert to project P_2, which saves the cost of hiring new expert. Similarly, we hire experts for project P_3 as well. When we apply Cluster Hire problem to social networks then we have to consider the communication cost of an expert as well while hiring the experts. So, our goal will be to find a collaborative group of experts with minimum communication cost, who can complete given three projects within the given budget and working capacity. That would be an effective solution to the Cluster Hire problem in social network.

2.1.3 Difference between TFP and Cluster Hire

Given a set of n available experts, each associated with some skills, and given a set of m projects of an organization that requires some people with specific skills. Figure 3 shows an example of the Cluster Hire problem. It explains the main components of the Cluster Hire problem and depicts a feasible solution to the problem.

In Cluster Hire problem, when a group of experts is hired for *a set of projects*, certain factors are kept in mind. These factors are mentioned below:

- The required skillset of projects
- The budget for completing projects
- The communication cost between the hired experts
- The workload capacity of the experts
- The profit extracted after the successful completion of each project

Whereas Team Formation problem refers to hiring the efficient team of experts for the completion of one *single project*. When hiring for a single project, the main factors to keep in mind are the required skillset of a project, Budget for a project and communication cost between experts. So, there are far less factors to consider while hiring, which makes Team Formation problem simple compared to Cluster Hire problem.

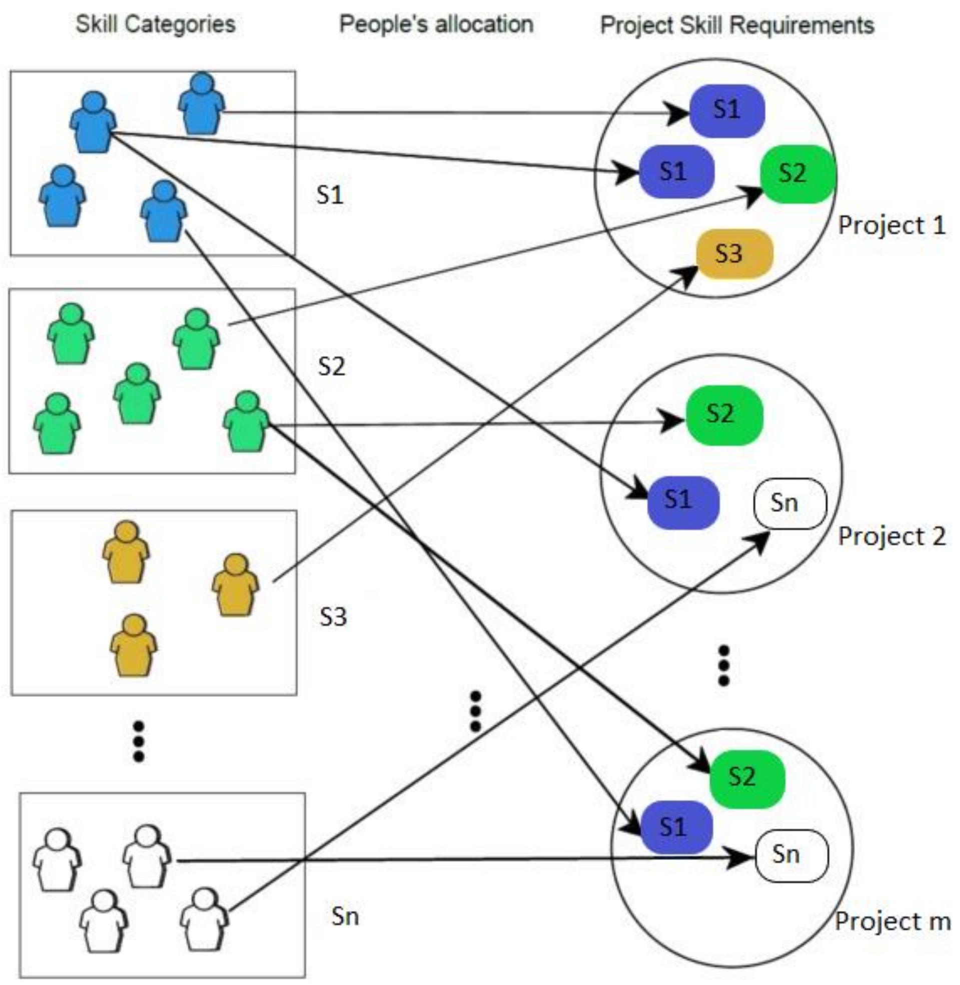


Figure 3: Cluster Hire Example

In the Figure 3, we have *multiple projects* on the right-hand side; each of them has a set of required skills. To complete these projects, for each needed skill, we must hire an expert who can fulfill that particular skill. For instance, Project 1 requires two experts for skill S_1 and one expert each respectively for skill S_2 and S_3. As per Figure 3, two experts from having skill S_1 are allocated to Project 1. One of the two experts hired for skill S_1 works for Project 1 and Project 2 at the same time, which saves hiring cost for the organization. This acts as a good example of a Cluster Hire problem.

However, if we pick just a single project from Figure 3 and hire a team of experts for *a single project*, then it is called the Team Formation problem. For example, project 2 requires one expert each for skills S_1, S_2 and S_n. So, we hire a team of experts without worrying about working capacity of the experts or profit of the project. This is an example of a Team formation problem, where you hire a team of experts for a single project.

Our thesis is focused on solving Cluster Hire problem in social networks. While hiring the group of experts for the set of projects, we also make sure that the hired group of experts are collaborative i.e., they have less communication cost between them. Information about communication cost is stored in the social network, which is represented as a weighted graph. The weight of an edge between two experts represents value of communication cost. We will discuss more about social networks in following section.

2.2 Fundamental Concepts

2.2.1 Social Networks

John Scott described Social Networks as the most popular way to model the interactions among the people in a group or community [1]. Social networks can be represented as graphs, where a vertex corresponds to an expert/person in some group, and an edge represents a form of relationship/association between the corresponding two experts [1]. The interests which are intrinsic to any community drive the associations in social networks. John Scott in [1] describes Social Networks as a set of nodes tied together by the set of relations(edges) between them, and these social networks follow a complex pattern and form a complex system of vertices and connections (edges) between them. A social network can be represented as a weighted or unweighted graph [1]. A social network graph can be represented as weighted graph, which is denoted by G = (V, E, W), where V is the set of vertices (actors), E is the set of edges (relationships), and W is the weight of the edges (relationships) [1].

Figure 4 displayed below is an example of a social network graph, where each node represents expert and edges connecting two nodes has weight associated with them. In this case, edge weight represents communication cost between two experts and each expert has some skills, which are represented in Figure 4.

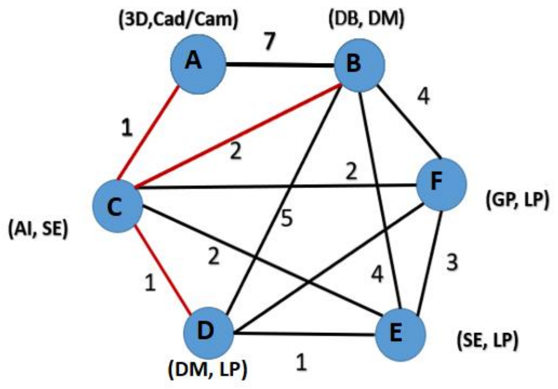


Figure 4: Social network graph

Moreover, the author writes that social networks are ubiquitous and can be created from various disciplines such as Sociology, Twitter friendship, LinkedIn, protein network, etc. [1]. The Social Network is a large graph, unlike other graphs. Despite the similar appearance of the networks, there are some characteristic features of the Social Network, such as degree of distribution, clustered coefficient, and path distance (6 degrees of separation) [1].

Several relational and structural approaches to social analysis have been developed to form social networks. However, ideas of culture and cultural formation and classical sociology were applied to demonstrate social patterns, social behavior, and social feelings. All the factors mentioned above-described social networks in the past. Additionally, there is another approach which only focuses on the actual patterns of interaction and interconnection by which the social groups and community are interlinked. In some instances, it is defined as a social system or a social organism. In 1929, Frigyes Karinthy was the first person in mankind to introduce the concept of 6-degrees of separation, which means that any two randomly selected people in the world are 6-steps away from each other. Later, in 1960, Stanley Milgram experimented to find the average path length between those individuals, and it came out to be 5.9. Another prominent characteristic of the Social Network is Degree Distribution. It is defined as the probability of the number of connections of a node with other nodes over the complete network. In 1999, Albert-Lszl Barabsi et al. said that some nodes had many more connections than others, called the hub, and they used the term "scale-free network" [2].

2.2.2 Importance of Social Network for efficient team formation

A social network that is considered for team formation to accomplish complex tasks is directly associated with effective team organization. Matthew et al. [24] conducted several experiments and concluded that such social structures are vital for task completions and successful completion of any project work. There is a subgraph associated with each project and each team, where the set of vertices represents the experts who have the collective skills required for the fulfillment of the tasks involved. They collectively run experiments to study the network effects on real-world data, representing how social networks amongst the team members can affect team performance and dynamics. Lars et al. [25] studied the dynamics of group formation where they considered three parameters, namely membership, growth, and evolution, which helped them to analyze the evolvement of large groups in an organization/community. They concluded that multiple factors influenced an individual/expert's desire and willingness to join a specific community/organization.

Additionally, they performed research on these factors to analyze their scalability. A previous study shows the correlation between a person's willingness to join a community and the number of individuals he is friends with within that organization/community. There are multiple techniques used to identify the most prominent determinants influencing such a social group; these techniques are known as decision-tree techniques.

A novel methodology is further devised that measures the incoming and outgoing of individuals from one community to another based on their level of interest in a particular/specific topic. The DBLP dataset is used to extract the results from experiments and methodologies.

2.2.3 Social Network Analysis (SNA)

Social Networks Analysis (SNA) can be simply defined as the in-depth analysis of social network structure, the tendency towards the time, the pattern of a relationship with social actors, and the available data along with them [1]. Since social networks are formed mostly with our environmental structure, researching its primary measures such as closeness centrality, betweenness centrality, degree centrality, diameter, etc., will provide more robust results, which would be an innovative change in the world [1]. To analyze a social network, we need to convert it into a graph with nodes and edges, where nodes are social actors (can be a person, organization, or any other), and edges are the relationship between them [1]. The graph can either be weighted or unweighted (Weight mostly decided based on the similarity of two nodes, the distance between them, or the frequency of collaborations) [1]. At the same time, it can be either directed or undirected, too [1]. Therefore, social network analysis uses the concepts of graph theory.

Another unique characteristic of the social network is its dynamic behavior. However, real-world social networks are unsteady and not static, such as LinkedIn, Facebook, Twitter, Instagram, Snapchat, etc. They possess quick expansion and evolve quickly over time. These unpredictable changes make social networks complex and very difficult to understand, and even more challenging to tackle.

In the present-day world, Social Network Analysis is applied to various businesses and disciplines such as academics, politics, healthcare, and daily life activities [1]. It is mostly applied to help improve the effectiveness and efficiency of decision-making processes [1]. These kinds of applications are used in Law Enforcement agencies, Civil Organizations, Health Care, and Social Networking Sites like Facebook, Instagram, Snapchat, and LinkedIn. Some applications of social network analysis are to help analyze the spread of diseases [3], to detect terrorist activities [4], to observe dynamic co-authorship networks [5], and much more research applications in the real world.

2.2.4 Various SNA Problems

According to John Scott [1], Social Network Analysis deals with different issues. They are as mentioned below:

- Team Formation Problem
- Link Prediction
- Community Detection
- Collaborative Recommendation
- Leadership Detection
- Migration Between Communities
- Sentimental Analysis
- Insurance Analysis

- Fraud Detection

We are going to discuss some of the above problems here. The Team Formation Problem (TFP) refers to hiring a team of experts to complete an assigned project, which requires a set of skills to be accomplished, where a set of skills needed is a subset of a set of the total number of skills. Link prediction predicts missing links in current networks and new or dissolution links in future networks [7]. Community detection can be defined as finding people with similar tastes, choices, and preferences to get associated with a social network that leads to the formation of virtual clusters or communities [8]. Collaborative Recommendation identifies users whose tastes are similar to those of the given user and recommends items they have liked [9].

In our research, the aim is to solve the problem of Cluster Hire (which is an extension of TFP) in social networks by minimizing the communication cost. Hiring a group of experts with a specific skillset for a set of projects is a difficult task. Finding a group of experts in real life from hundreds of people is a very costly and not feasible task. Therefore, we aim to solve the cluster hire problem in social networks consisting of many experts. We implement our solution using a Modified Weighted Structural Clustering Algorithm for Networks (MWSCAN), which is a clustering algorithm. We will discuss some clustering concepts and approaches in the following section.

2.2.5 Clustering

Grouping similar elements together in a group is defined as clustering. With clustering, many groups can be formed based on their characteristics and considerations. Graph clustering is a special type of clustering, and it is used on large graphs or networks. Moreover, Network clustering (graph partitioning) is an important task for the discovery of underlying structures in networks and helps with social network analysis [10].

There are various categories in clustering. They are as mentioned below:

- Hierarchical based clustering
- Density-based clustering
- Partitioning based clustering
- Grid-based clustering

Density-based clustering

Density-based Clustering relies on the nodes in the graph for grouping. A cluster is formed by the vertices, which are highly connected. Some examples are DBSCAN (Density-Based Spatial Clustering of Applications with Noise), AHSCAN (Agglomerative Hierarchical Structural Clustering Algorithm for Networks), DHSCAN (Divisive Hierarchical Structural Clustering Algorithm for Networks), SCAN (Structural Clustering Algorithm for Networks). DBSCAN algorithm was proposed for spatial clustering data with noise. Because of its unique features, this algorithm became rapidly famous in various fields. Application of this algorithm includes in the field of science such as grouping spatial civil infrastructure network, chemistry, spectroscopy, medical diagnosis based on medical images (brain atrophy, skin lesions), and social science (pheromone data) [11]. DBSCAN is applied to segregate the 3D images in remote sensing as well.

The disadvantage of DBSCAN is that it is unreliable when the border elements of the two clusters are relatively too close. To counter this, Xiaowei Xu et al. [10] proposed the Structural Clustering Algorithm for Networks (SCAN). This algorithm was capable of making dense clusters as well as efficiently handle the weakly connected nodes to hubs. These hubs were described as those nodes which are interlinked to two or more clusters at a time.

2.2.6 Unweighted graph clustering with SCAN

Xiaowei Xu et al. [10] proposed a new method to cluster a network graph based on structural similarity. This was used to cluster the undirected graph as well as an unweighted graph based on structural similarity. Therefore, the author named it a Structural Clustering Algorithm for Networks (SCAN). It detects clusters, hubs, and outliers by using the structure and the connectivity of the vertices as clustering criteria [10]. This algorithm finds the Core node out of the network. The Core node is chosen based on the number of neighbors in its neighborhood that are structurally similar. Later, by considering this Core node as the seed for the cluster, it builds up a cluster around it. This approach divides the graph into three parts: Clusters, Hubs, and Outliers, which are illustrated in the following Figure 5. Hubs are essentially nodes that bridge many clusters, and outliers are nodes which are marginally connected to cluster featuring weak connection. In Figure 5, there is one hub that connects two clusters and works as a bridge between two clusters. Also, we have one outlier, which is weakly connected to one of the clusters.

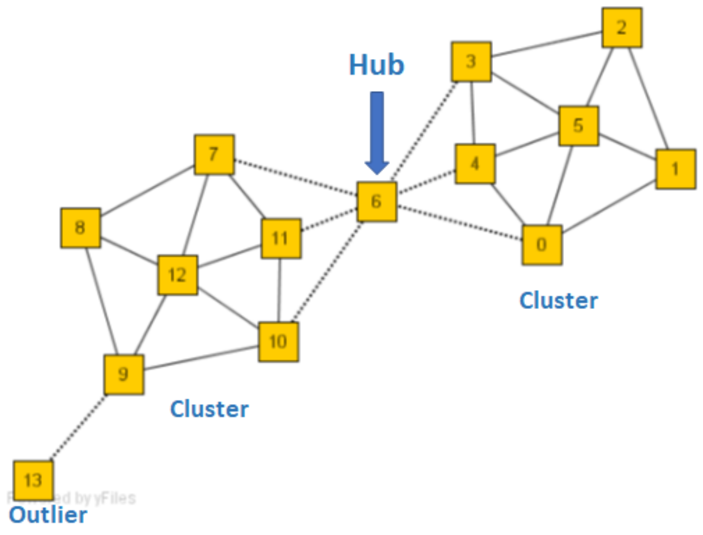


Figure 5: Working of SCAN [10].

In this research paper [10], they focused on the simple unweighted and undirected graph. Let $G(V, E)$ be a graph, where V is a set of vertices or nodes and E is a set of edges connecting two vertices. Structure of a vertex is defined by its neighborhood. Formal definitions from the research paper [10] are described below,

Definition 1. (Vertex Structure)

Let $v, w \in V$, the structure of v is defined by its neighborhood. Vertex structure of vertex v is denoted by $\tau(v)$ [10]

$$\tau(v) = \{w \in V \vee (v, w) \in E\} \cup \{v\}$$

Definition 2. (Structural Similarity)

Structural similarity between two vertices v and w will be large if they share a similar structure of neighbors that is a frequent regime of working together and their structural similarity is denoted by $\sigma(v, w)$ [10]

$$\sigma(v, w) = \frac{|\tau(v) \cap \tau(w)|}{\sqrt{|(\tau(v)||\tau(w))|}}$$

They normalize the number of common neighbors of the two vertices by the geometric mean of the two neighborhoods' size. Because if we only use the number of shared neighbors, then vertex 6 in Figure 5 will be clustered into either of the clusters or cause the two clusters to merge [10].

Definition 3. (ε - Neighborhood)

For vertices v and w, we apply threshold ε to structural similarity $\sigma(v, w)$, when assigning cluster membership to vertex v [10]. That is called the ε – Neighborhood for vertex v and is denoted by $N_\varepsilon(v)$

$$N_\varepsilon(v) = \{w \in \tau(v) \mid \sigma(v, w) \geq \varepsilon\}$$

Definition 4. (Core)

A vertex $v \in V$ is called a core with reference to ε and μ , if its neighborhood contains at least μ vertices, where μ is number of neighborhood experts connected to core vertex. So, core expert is the highly connected expert [10]. The core vertex is denoted by $Core_{\varepsilon,\mu}(v)$, where following condition is fulfilled,

$$|N_\varepsilon(v)| \geq \mu.$$

Definition 5. (Direct Structure Reachability)

We grow clusters from the core vertex as follows. If a vertex is in ε – Neighborhood of a core, it should also be in the same cluster. They share a similar structure and are connected. This idea is represented in the below formula [10]

$$DirREACH_{\varepsilon,\mu}(v, w) = Core_{\varepsilon,\mu}(v) \wedge w \in N_\varepsilon(v)$$

The Pseudo Code of the Algorithm SCAN [10]

```
ALGORITHM SCAN ($G(V, E)$, $\varepsilon$, $\mu$)
// all vertices in $V$ are labeled as unclassified;
for each unclassified vertex $v \in V$ do
// STEP 1. check whether $v$ is a core;
    if $Core_{\varepsilon,\mu}(v)$ then
```

```
// STEP 2.1. if $v$ is a core, a new cluster is expanded;
        generate new clusterID;
        insert all $x \in N_{\varepsilon}(v)$ into queue $Q$;
        **while** $Q \neq 0$ **do**
          $y$ = first vertex in $Q$;
          $R = \{x \in V \mid DirREACH_{\varepsilon,\mu}(y, x)\}$;
          **for** each $x \in R$ **do**
            **if** $x$ is unclassified or non-member **then**
              assign current clusterID to $x$;
            **if** $x$ is unclassified **then**
              insert $x$ into queue $Q$;
          remove $y$ from $Q$;
  **else**
// STEP 2.2. if $v$ is not a core, it is labeled as non-member
      label $v$ as non-member;
**end for**.
// STEP 3. further classifies non-members
**for** each non-member vertex $v$ **do**
  **if** ( $\exists\, x, y \in \Gamma(v)$ ( $x$.clusterID $\neq$ $y$.clusterID) **then**
      label $v$ as hub
  **else**
      label $v$ as outlier;
**end for**.
**end SCAN.**
```

The SCAN algorithm starts by visiting each vertex once to find structure-connected clusters and then to identify hubs or outliers visiting the isolated vertices [10]. For given parameter settings, the SCAN algorithm performs one pass of a network and finds all connected clusters. In the beginning, all vertices are labeled as unclassified and entered into the queue. The SCAN algorithm classifies each vertex, either a member of a cluster or a non-member. For each unclassified vertex, SCAN checks whether that vertex is a core. If the vertex is a core, a new cluster is expanded from this vertex. Otherwise, the vertex is labeled as non-member [10]. To form a new cluster SCAN starts with the core vertex and find all the structurally reachable vertices from the core vertex. SCAN generates the new cluster ID, which will be assigned to each member of the cluster. To find the members of the cluster, it adds all unclassified neighborhood vertices of the core vertex in the queue. For each vertex in the queue, it finds all directly reachable vertices, which are unclassified, and inserts those vertices into the queue . These steps are repeated until the queue is empty [10]. The remaining non-member vertices can be labelled as hubs or outliers. If the isolated vertex has two or more edges with the other vertices, then it can be classified as a hub. Whereas other vertices will be classified as outliers [10].

2.2.7 Weighted graph clustering with WSCAN

Weighted graphs have a definite number associated with the edges, which is usually called edge weight [13]. The clustering technique discussed above in section 2.2.6 targets unweighted graphs. However, when provided with a weighted graph, the SCAN algorithm will simply ignore the edge weights and will perform clustering based on the structural properties of the graph. This might be acceptable in some instances, but sometimes, it is entirely inadmissible [14]. To overcome this problem, the author of [14] research paper and [13] thesis research work proposed a new algorithm called Weighted Structural Clustering Algorithm for Networks (WSCAN) as a solution to perform clustering in weighted graphs based on structural similarity. The algorithm presented slightly modified formula for structural similarity, which is defined as below,

Definition 6. (Extended Structural Similarity):
Let $G(V, E)$ be a graph, where V is a set of vertices or nodes and E is a set of edges connecting two vertices. Let $v, w \in V,$ and function below show the structural similarity between two vertices in the graph, where $w(v, w)$ represents the weight of an edge between experts/vertices v and w [13][14]. The structural similarity is denoted by $\sigma(v, w)$

$$\sigma(v, w) = \frac{|\tau(v) \cap \tau(w)|}{\sqrt{|(\tau(v)||\tau(w))|} \;\; w(v, w)}$$

In the above equation, the extended structural similarity of two vertices will be large if they share a similar structure of neighbors with communication cost between them low, which mean they have frequent regime of working together. It takes weight of edge into consideration when calculating structural similarity between them. WSCAN algorithm uses all the other formulas same as used in SCAN algorithm except for the extended structural similarity formula. WSCAN algorithm follows the same steps which are described in the pseudocode of SCAN in section 2.2.7 with the modified formula for the structural similarity.

2.3 Literature Review and Related Work

2.3.1 Team Communication Cost functions in social networks

The Team Formation problem was first introduced to the community of data mining and data management by Lappas et al. [6]. There were two functions proposed by them for estimating the value of the communication cost of a team. The first function measured the team communication cost by the diameter of the subgraph created by the hired team; diameter is measured by largest shortest path between any two nodes of the subgraph. The second function calculated the team communication cost by the cost of the minimum spanning tree (MST) on the subgraph. However, both the approaches had drawbacks as first approach only measures communication cost between two experts furthest away from each other in the team. Whereas second approach also does not measure communication cost of all the experts working in the team. Therefore, to counter those drawbacks, Kargar et al. [22] came out with two new team communication cost functions based on their experiments conducted.

These two functions are as described below:

- The first function is called *sumofdistance* function, which calculates the team communication cost based on summing up the shortest distance between all the experts who possess the necessary skills for the given project or in other words who are part of a team.

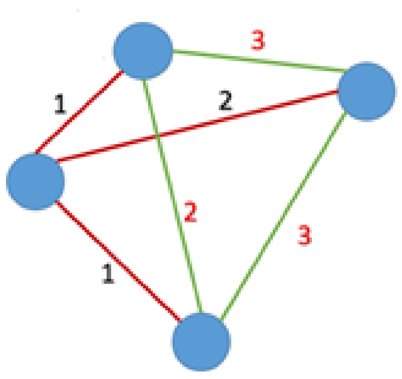


Figure 6: A weighted graph of team

In Figure 6, if we calculate the team communication cost, then it will be the sum of all the edge weight/distance between all vertices.

Team communication cost = 1 + 1 + 2 + 2 + 3 + 3 = 12

- The second function focuses on calculating the team communication cost where there exists a leader in the team. In this setup, the team of experts has a designated leader, who is responsible for the coordination and monitoring of the project. Each expert in the team has to communicate with the leader; therefore, communication cost of a leader is important while calculating team communication cost. This function considers the communication cost of the leader, along with all other experts of the team. It is known as the *leaderdistance* function, which discovers the sum of the shortest path between each expert and the leader in the project to calculate the team communication cost.

We used the *sumofdistance* function to calculate the communication cost among a selected group of experts because in our setup, there is no leader and *sumofdistance* considers all the experts in the hired group of experts.

2.3.2 Online Team formation problem in Social Networks

Anagnostopoulos et al. [30] conducted a study and experimented on the online version of the team formation problem. They used the social network of experts, which possesses a set of skills to carry out multiple projects and have compatibility and past collaboration with each other obtained from their prior work experience. They considered a sequence of tasks that arrive in an online fashion, each task requiring a specific skill set. The basic idea of the paper is to form multiple teams for multiple projects; each project and the associated set of tasks appears in an online fashion. The aim was to establish the team keeping in mind three attributes discussed below:

- The skills needed for project completion are covered
- The team formed is collaborative
- The working capacity of the individual is not overloaded

In the past, various heuristic and approximate algorithms have been used to solve the TFP problem with the above objectives and followed by results analysis and conclusions.

2.3.3 Clustering Technique for TFP in social network

Team Formation Problem (TFP) in a social network is popular and has been studied widely by many researchers and authors. Many techniques were proposed, and multiple experiments were conducted to solve the TFP, for discovering and identifying the experts that could cover all the tasks required for a project. They also kept in mind that the team formed is the most collaborative to reduce the time needed to carry out that project. With this objective, Kalyani et al. [21] in their paper, proposed the clustering technique, which groups a set of experts in the form of a cluster, where the cluster points represent experts with the required skills for the project. They have adopted the weighted SCAN (WSCAN) algorithm, which is an enhanced/extended version of SCAN, i.e., Structural Clustering Algorithm for Networks, to solve the team formation problem (TFP) with minimal communication cost. The communication cost is calculated based on past collaboration and work experience (worked under multiple projects before) between two hired experts. Their technique revolves around the basic idea of identifying clusters, hubs, and outliers in the network/graph of experts, where every node represents the expert, and the edge weights represent the communication cost. Firstly, a pool of experts was considered to approach the problem, who possess the skills required to carry a specific project. Followingly, they searched for the core, i.e., the highly connected expert among all experts. Then, the cluster is expanded from the core to the neighborhood nodes, which means from the densely connected nodes to the loosely connected nodes. The cluster expansion is carried out until the threshold range of communication cost is reached. Results comparison of Weighted SCAN for TFP is made with a greedy, genetic, random, and exact algorithm, and WSCAN produced faster results compared to all other algorithms [29].

2.3.4 Cluster Hire in social network using Greedy Algorithm

The Cluster Hire problem in social networks was first studied by Meet et al. [23]. Their research was bi-objective, where they hire a specific group of individuals that can extract the maximal profit and have minimum communication cost based on their past collaborations and work experiences. Specifically, they aimed to solve the Cluster Hire problem in a social network with the greedy algorithm. Their work was based on the assumption that there exists a social network among experts where each expert is associated with another expert through past work collaborations in one or more projects. Additionally, each expert in the formed setup is associated with a salary, i.e., the hiring cost. Also, experts have their working capacity. The hiring is done by keeping in mind that the sum of the wages of all the hired experts working for the projects would not exceed the budget allotted for the specific set of projects. Moreover, it is taken into account that no experts in the team formed are overloaded or working more than their limit or work capacity.

Two greedy algorithms were proposed to solve the bi-objective problem, namely, Expert Pick Strategy and Project Pick Strategy, to achieve all the attributes mentioned above during the team selection. Both of these algorithms used the scoring functions which are described below [23].

Table 2: Notations used for greedy algorithm [23]

E	Set of experts
P	Set of projects
$C(e)$	Cost of an expert e
$PF(p)$	profit of completing project p
$Cap(e)$	capacity of an expert e
$Dist(e, e')$	communication cost between two experts e and e'
λ	Trade-off parameter between profit and communication cost

ε	Final group of experts

The Expert Pick strategy picks an expert in each iteration based on their score and that expert is added in the final group of experts. Here is the scoring function they used to assign scores to each pair of experts and projects, which is denoted by sc_e^p.

$$sc_e^p \leftarrow \lambda . \frac{PF(p). \min \{Skill(e,p), Cap(e)\}}{C(e)} + (1-\lambda). \frac{a}{\sum_{e' \in \varepsilon} Dist(e,e')}$$

Where, $Skill(e,p)$ represents the number of skills in project p that could be covered by an expert e and a is just normalization constant. You can find information of all the notations in Table 2 above. The Expert Pick Strategy works by scoring each expert in the social network who can cover one or more required skills needed for the successful completion of the projects in a way that it covers numerous skills of the high-profit project and communication cost between experts remains low. Between the number of skills that expert e can cover in p and the capacity of e, we choose the minimum value to make sure we do not exceed the capacity of an expert. The cost of an expert is also considered while scoring to make sure that hiring is done within the budget. In each iteration, they assign score to pair of experts/projects and choose the expert with the highest score.

Since there are negligible chances that the hired expert is cheap and can cover many skills along with minimal collaboration, they introduced λ as a trade-off parameter between profit and communication cost, which had a value between 0 and 1. This trade-off is beneficial in determining and calculating whether to put more weight on communication cost or profit margin.

Talking about the second strategy, the Project Pick Strategy chooses a project in each iteration by assigning scores to uncovered projects using their scoring function. The goal is to complete the projects with a maximum profit margin, the set of experts hired to accomplish the projects costs less or within the budget and can communicate efficiently with a background of past collaborations and work experiences. In each iteration, first they find teams for each uncovered project and after that they use a scoring function to score these projects. A project with highest score is chosen in each iteration.

In each iteration, and for any uncovered project, they find a set of experts to cover that project. To find this set of experts for project p, they start with an empty set E_p. After that, they select an expert to be added to E_p that maximizes the following scoring function [23],

$$sc_e \leftarrow \lambda . \frac{\min\{Skill(e,p), Cap(e)\}}{C(e)} + (1-\lambda) . \frac{a}{\sum_{e' \in E_p} Dist(e,e')}$$

Once, we have a set of experts E_p for all uncovered projects, next step is to score the projects using following scoring function and highest scoring project will be added to the pool of selected project in each iteration [23].

$$\lambda . \frac{PF(p)}{\sum_{e \in E_p} C(e)} + (1-\lambda) . \frac{b}{\sum_{e \in E_p} \sum_{e' \in \varepsilon} Dist(e,e')}$$

All the notations used in the equations are described in Table 2 above. a & b used in the equations are just normalization constants. They further ran their experiments on the real dataset and proved the efficiency of their approach using the greedy algorithm when compared with the random algorithm used in social networks.

CHAPTER 3
PROPOSED METHODOLOGY

3.1 Problem Statement

Let $E = \{e_1, e_2, \ldots., e_n\}$ denote a set of n experts; $S = \{s_1, s_2, \ldots., s_m\}$ denote a set of m skills (All symbols used in the thesis are summarized in Table 3 below). Each expert e posses a set of skills, denoted by $ES(e)$. Each expert e demands a salary, i.e., hiring cost to participate in projects, which is indicated by $C(e)$. The hiring cost of each expert is predetermined, which remains unchanged irrespective of the number of projects assigned to experts or the number of skills experts have. A set of given projects is denoted by $P = \{p_1, p_2, \ldots., p_k\}$. Each project is also composed of a set of required skills that need to be covered by experts for the completion of a project. The set of skills required for project p is denoted by $PS(p)$. Each expert e has a working capacity denoted by $Cap(e)$, Which is the maximum number of projects an expert can participate in at the same time. Working capacity is to make sure that we do not overload an expert with too many projects. One expert can participate in multiple projects but can not fulfill more than one skill for the same project. We are also given a budget B to hire a group of experts for a set of projects.

Definition 1. Group of Experts: Given all of the above, a group of experts $\mathcal{E} \subseteq E$ can complete a subset of projects $\mathcal{P} \subseteq P$, if the following holds:

1. $\forall\, p \in \mathcal{P}$ and $\forall\, s \in PS(p)$, an expert e in $\mathcal{E}$ is assigned to perform the required skill s for p.

2. $\forall\, e \in \mathcal{E}$, e is not covering more projects than $Cap(e)$.

3. $\forall\, e \in \mathcal{E}$, $\sum_{e \in \mathcal{E}} \mathrm{C}(e) \leq B$, which means the sum of the hiring cost of each expert e in $\mathcal{E}$ is within the budget B.

4. The total communication cost $CC(\mathcal{E})$ is minimal, meaning they have high collaboration among experts.

Table 3: Symbols used in the Thesis

Symbols	Notation Definition
E	Set of n experts
S	Set of m skills
P	Set of k projects
$ES(e)$	Set of skills possessed by expert e
$PS(p)$	Set of required skills by project p
$C(e)$	Cost of hiring an expert e
$PF(p)$	Profit of completing a project p
$Cap(e)$	Capacity of an expert e
e_i	i^{th} expert
G	A social network graph of n nodes
$Dist(e_i, e_j)$	Distance/Communication cost between e_i and e_j in G
$CC(\mathcal{E})$	Communication cost among a group of experts $\mathcal{E}$
B	Total budget for hiring group of experts for a set of projects

There exists a social network modeled as an undirected weighted graph G, where the experts are connected to each other. Each expert e_i is denoted as a node of graph G. The terms node and expert are used interchangeably in this work. Two experts in a graph are connected to each other via edge if they have the previous collaboration, i.e., working on the same projects in the past. Edge weight denotes the communication cost between these two experts. The smaller the edge weight, the stronger the strength of collaboration between two experts. When there exists no direct collaboration between two experts, the communication cost between them is determined by the weight of the shortest path between two experts in graph G. If the expert is not connected to any other expert in a graph, we consider the communication cost value for a given expert as infinity.

We aim to hire a group of experts, where communication cost among them is minimal. The communication cost among a group of experts is calculated as the sum of distances between each pair of experts in the group, which is known as the *sumofdistance* function [22]. In Figure 7, we have a graph of experts on the left-hand side, and we display hired group of experts for the required skillset on the right-hand side. The graph of experts in Figure 7 represents experts as nodes, weighted edges between experts represent communication cost, and each expert's skills are also shown.

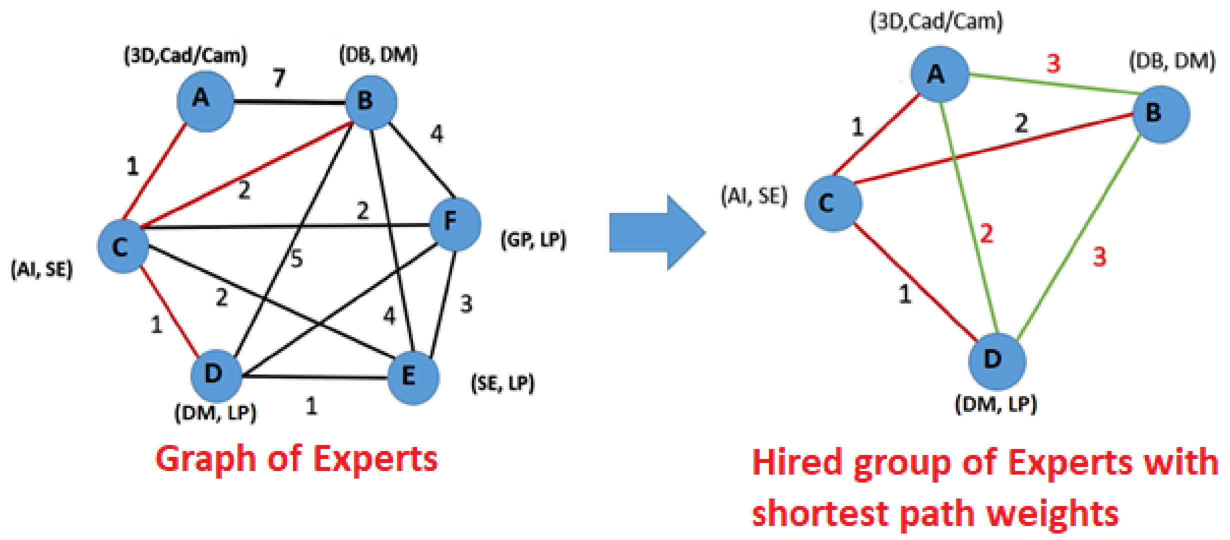


Figure 7: Shortest path among experts

In the hired group of experts, we use the weight of the shortest path between experts as communication cost between experts when there exists no edge between experts in the graph of experts. For example, experts A and D do not have a connecting edge in the graph of experts in Figure 7. So, we find the shortest path between A and D, which is from A to C and C to D. We make the summation of respective edge weights (1+1=2) to find the communication cost value between A and D. Therefore, we display a weighted edge between experts A and D with the weight of "2" in Figure 7. The shortest path can also be used to minimize the communication cost between experts connected to each other in the graph. For example, in Figure 7, experts A and B have a weighted edge between them with the weight of "7" in the graph of experts, but there exists a shortest path between A and B via C, and the weight of this shortest path is the sum of weights between A and C and then C and B (1+2=3). Therefore, we change the weight of an edge between A and B from "7" to "3" in the graph on the right-hand side in Figure 7. This way, we find the shortest path between each expert in a hired group of experts. These shortest paths between experts are displayed in green color, and the shortest path weights are displayed in red color in Figure 7.

Communication cost among the hired group of experts is calculated by the *sumofdistance* approach. We have hired a group of experts with edge weight between each pair of experts in the group in Figure 7. So, we perform the summation of distances between each pair of experts to find the group communication cost. For example,

Required skillset = {DB, AI, 3D, LP}
Hired group of experts = {B, C, A, D}
Group Communication Cost = (1+2+3+2+1+3) = 12

3.2 Proposed Strategy: MWSCAN

3.2.1 Definitions related to MWSCAN

We will discuss some important terminologies related to Modified Weighted Structural Clustering Algorithm for Networks (MWSCAN).

1. Graph of experts: It can be defined as G(V, E, W), where,
   Vertices V: refers to a set of experts (E).
   Edges E: a set of edges between experts, representing connections among experts.
   Edge weights W: a set of edge weights between experts, representing communication cost between experts.

2. Communication Cost: It can be defined as a distance/weight between two experts e_i and e_j on a graph G. It is denoted by $Dist(e_i, e_j)$. When there exists no direct connection between experts, the weight of the shortest path between two experts is considered as a communication cost between them.

3. Sum of Distances: The communication cost of a group of experts is the sum of distances between each pair of experts in a group, which is defined as

$$\textit{sumofdistance} = \sum_{i=1}^{x} \sum_{j=i+1}^{x} Dist(e_i, e_j)$$

   Here, x is the number of experts in the group of experts[21].

4. Vertex Structure: Let $e_i \in E$ (set of experts), the vertex structure of e_i is defined by its neighborhood, denoted by $\tau(e_i)$ [21].

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

Here, (e_i, e_j) represents the edge between experts e_i and e_j, which belongs to a set of edges E in graph G.

5. Extended Structural Similarity: Structural similarity between two vertices e_i and e_j will be large if they share a similar structure of neighbors. It means that they have a high frequency of working together, and communication cost between them will be low. Structural similarity between experts e_i and e_j is denoted by $\sigma(e_i, e_j)$ [21].

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

Here, $\sigma(e_i, e_j)$ is inversely proportional to communication cost. If $\sigma(e_i, e_j)$ is high, then it implies that communication cost between experts e_i and e_j is low.

$$\sigma(e_i, e_j) \propto \frac{1}{Dist(e_i, e_j)}$$

Suppose if e_i and e_j have high communication cost between them, then the frequency of working together is low and vice versa.

6. Neighborhood (ε): We apply threshold ε to structural similarity when assigning cluster membership to expert e_i and by applying threshold ε, we can define expert's ε-neighborhood, which is denoted by $N_\varepsilon(e_i)$ [21].

$$N_\varepsilon(e_i) = \{e_j \in \tau(e_i) \mid \sigma(e_i, e_j) \geq \varepsilon\}$$

7. Core Expert: A vertex $e_i \in E$ is called a core with reference to ε and μ , if its neighborhood contains at least μ vertices [21]. The core expert is denoted by $Core_{(\epsilon,\mu)}(e_i)$, if it satisfies the following condition,

$$|N_\varepsilon(e_i)| \geq \ \mu$$

Where μ is the number of neighborhood experts connected to core vertex (highly connected expert).

3.2.2 MWSCAN algorithm

MWSCAN algorithm is our solution to the Cluster Hire problem in social networks. This algorithm receives a set of n experts and set of k projects, a social network graph G, and assigned budget B as an input. In our setting, each expert has skills, working capacity, and cost of hiring, and each project has its set of required skills. The output of the algorithm will be a group of experts $\mathcal{E}$ who can cover the required skills of a subset of projects $\mathcal{P}$ while minimizing the communication cost between a group of experts $\mathcal{E}$. The hiring of experts is done within the assigned budget, so the cost of hiring a group of experts does not exceed the budget assigned B.

Algorithm: Cluster Hire in social network using MWSCAN

Input: Social network graph G(V, E, W), Set of n experts $E = \{e_1, e_2, \dots, e_n\}$, Set of k projects $P = \{p_1, p_2, \dots, p_k\}$, each expert e_i has skills $ES(e_i)$, working capacity $Cap(e_i)$, and hiring cost $C(e_i)$ associated with it. Each project p_i has a set of required skills $PS(p_i)$. We also have Budget B assigned for a set of projects P.

Output: A group of experts $\mathcal{E} \subseteq E$ for a subset of projects $\mathcal{P} \subseteq P$, where $CC(\mathcal{E})$ is minimal.

```
Start
  1. Find Pool of Experts (PoE) from $E$
  2. All vertices are unclassified
  3. For each unclassified vertex $e_i \in E$;
       3.1 Count the number of experts in $\varepsilon$-neighbourhood of expert $e_i$
     End for
  4. The expert $e_i$ with the highest number of connections in $N_\varepsilon(e_i)$ will be the core expert
       4.1 If more than one expert has same number of experts in $N_\varepsilon(e_i)$, then
             4.1.1  Expert $e_i$, who covers more number of skills for $P$ within $Cap(e_i)$, will be
                    the core expert
             4.1.2  If experts cover same number of skills, then
```

```
                4.1.2.1 Choose core expert randomly out of the experts with highest number
                        of connections in $N_\varepsilon(e_i)$
5. Total_cost += $C(e_i)$
6. **If** Total_cost ≤ $B$, **then**
    6.1 Generate a new cluster from the core expert $e_i$
    6.2 Insert core expert $e_i$ into final group of experts $\mathcal{E}$
    6.3 Remove skills fulfilled by $e_i$ from the required skill set based on $Cap(e_i)$
7. **For** each $e_j \in N_\varepsilon(e_i)$;
    7.1 Obtain $Dist(e_i, e_j)$ from core expert $e_i$
   **End for**
8. **While** (required skillset is not empty AND Total_cost ≤ $B$) **do**
    8.1 Choose expert $e_j$ with minimum value of $Dist(e_i, e_j)$
    8.2 **If** more than one expert has same $Dist(e_i, e_j)$, **then**
        8.2.1 Choose expert $e_j$, who covers more number of skills for $P$ within $Cap(e_i)$
        8.2.2 **If** experts cover same number of skills, **then**
                8.2.2.1 choose an expert randomly out of the experts with minimum value
                        of $Dist(e_i, e_j)$
    8.3 **If** $e_j$ covers any skills in the required skillset within $Cap(e_j)$, **then**
        8.3.1 Total_cost += $C(e_j)$
        8.3.2 **If** Total_cost ≤ $B$, **then**
                8.3.2.1 Add expert $e_j$ into final group of experts $\mathcal{E}$
                8.3.2.2 Remove skills fulfilled by $e_j$ from the required skillset based on
                        $Cap(e_j)$
9. **Return** group of experts $\mathcal{E}$
10. **End**
```

The algorithm starts by finding the Pool of Experts (PoE) from the given set of experts E by removing all the experts who do not provide any skills from the required skill set of the projects. In the given graph G, all the vertices are unclassified at the start. We start by scanning each vertex and counting the number of neighborhood vertices, which exist in the ε-neighbourhood. An expert with the highest number of experts in its ε-neighbourhood will be our core expert. In the case of more than one expert having the same number of connections in their neighborhood, we select the core expert covering more number of skills in the required skill set within their working capacity. Once we find the core expert, we start forming a cluster around the core expert with the threshold of ε. All the experts present in ε-neighborhood of core expert will be part of the cluster. We also add core expert to the final group of experts $\mathcal{E}$ and remove the skills fulfilled by core expert from the required skillset. We add the hiring cost of core expert in the total cost.

For the remainder of the required skillset, we start scanning experts in the ε-neighborhood of the core expert and obtain the neighbor's distance/communication cost from the core expert. We first choose the neighbor expert with minimum from the core expert and check if chosen expert provides any skills in the remaining required skillset. If they provide any skills in the required skillset then we add the hiring cost of an expert to the total cost. Once we make sure that the total cost is within budget, we subsequently remove the respective skills fulfilled by an expert from the required skillset while keeping the expert's working capacity in mind. The selected expert is added to the final group of experts $\mathcal{E}$. We follow the same process until we have found experts for all the required skills or till the assigned budget is exhausted. In the end, we will get the final group of experts $\mathcal{E}$ who is hired to complete the given set of projects with minimum communication cost among the group of experts.

Example: We will be discussing a small example of the Cluster Hire problem in social networks and its solution using our proposed approach of MWSCAN.

Table 4: Set of Projects

Projects	**Required Skills: Required number of experts**	**Profit**	**Budget**
P1	Java: 2 Network: 1	50000	50000
P2	Report: 1 Data: 2 Algorithm: 1	35000	

Table 5: Set of Experts

Expert ID	Expert Skills	Cost / Salary	Capacity
A	Java, Data	5000	5
B	Report, Analytics, Network	7000	2
C	Security, Big Data	8000	3
D	Report, Data, Cloud	4500	4
E	Java, C++, Data	10000	4
F	SE, Algorithm, PM	7500	3
G	Cloud, Testing	6000	5
H	DB, Python, Security	3500	1

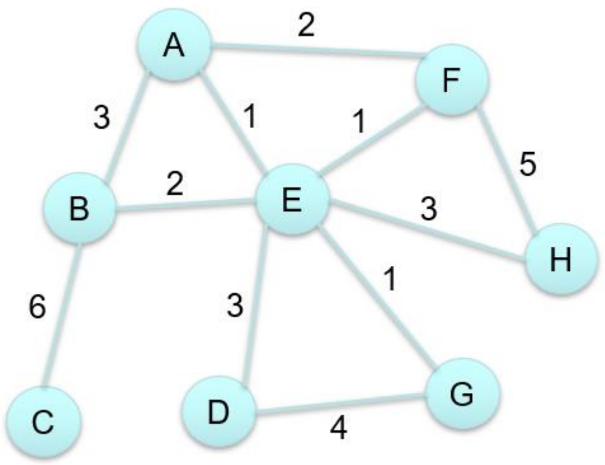


Figure 8: Graph of Experts

Problem: For the given set of projects and the given set of experts in Tables 4 and 5, we have to find a group of experts who can fulfill the requirements of the projects within the budget of 50,000 while keeping the communication cost low. Please note that Figure 8 is a graph of experts representing communication cost values between experts as edge weights.

Solution:

Step 1: Find a pool of experts (PoE) by removing all experts, who cannot fulfill any required skills of the given projects. Thus, we eliminate experts C and G from the Figure 8 to get Pool of Experts (PoE). The Pool of Experts for the given scenario is represented in Figure 9 below.

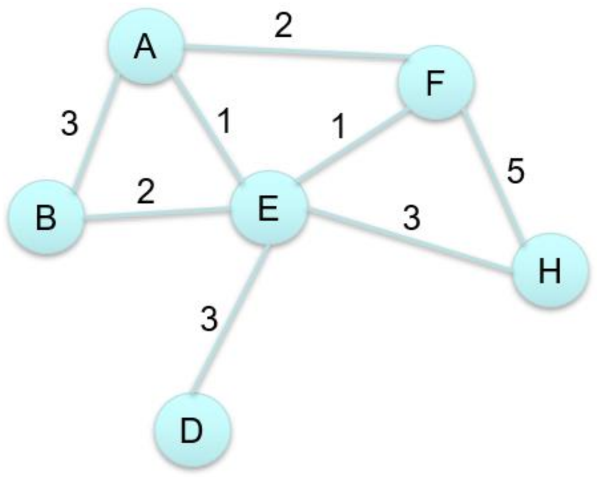


Figure 9: Pool of Experts (PoE)

Step 2: Next step is to find the core expert out of Pool of Experts. The Core expert will be an expert, who has maximum number of experts in its ε- neighborhood. In our scenario, expert E has the highest number of experts (5) in its ε- neighborhood. Therefore, expert E is the core expert out of given set of experts.

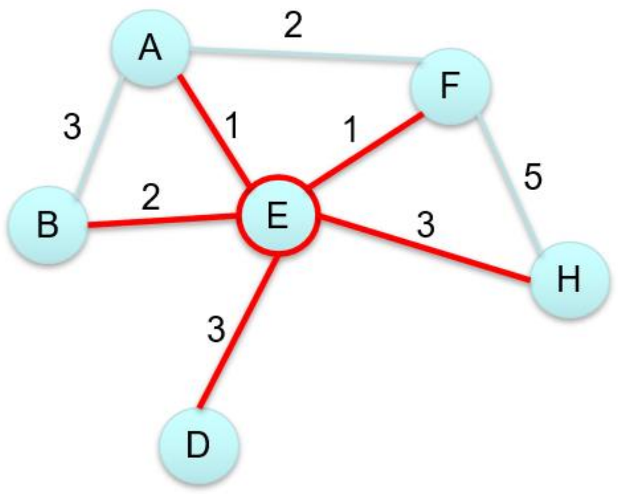


Figure 10: The Core Expert E

Step 3: After making sure that the expert E has a hiring cost is within the assigned budget of 50,000, we add the core expert to the final group of experts. We also remove the respective skills fulfilled by the core expert and the updated required skills for the projects are represented in Table 6 below. The skills fulfilled by hired expert at each step are highlighted in red color in Tables 6, 7, 8, 9.

Budget remaining: 50000 – Cost of an expert E = 50000 – 10000 = 40000

Group of experts: {E}

Table 6: Step 3 - Remaining projects' skills

Projects	**Required Skills: Required number of experts**	**Profit**
P1	Java: 1 Network: 1	50000
P2	Report: 1 Data: 1 Algorithm: 1	35000

Step 4: For the remaining required skills, we search for experts in the neighborhood of the core expert, by picking the expert with the least communication cost first. Here, Experts A and F both have communication cost of 1. In this case, we pick an expert who can fulfill a greater number of skills in the required skills of the projects while keeping their working capacity in mind. In this example, Expert A can cover two skills (P1-Java, P2-Data) and have a working capacity of 5. So, it can fulfill two skills for two projects. Whereas Expert F can cover only one skill (P2-Algorithm) and that is within its working capacity (3). Therefore, we hire Expert A and add it to the final group of experts for given projects requirements.

Budget remaining: 40000 – Cost of an expert A = 40000 – 5000 = 35000

Group of experts: {E, A}

Table 7: Step 4 - Remaining projects' skills

Projects	**Required Skills: Required number of experts**	**Profit**

P1	Java: 0 Network: 1	50000
P2	Report: 1 Data: 0 Algorithm: 1	35000

Step 5: We will iterate through all the neighborhood experts of core expert until we fulfill all the required skills of the projects, or we have exhausted the budget assigned for hiring experts for the given set of projects. As we pick the expert with the minimum communication first, we will pick an expert F, which fulfills one skill (Algorithm) in project P2. Thus, expert F will be added to the group of experts after checking for the budget.

Budget remaining: 35000 – Cost of an expert F = 35000 – 10000 = 25000

Group of experts: {E, A, F}

Table 8: Step 5 - Remaining projects' skills

Projects	**Required Skills: Required number of experts**	**Profit**
P1	Java: 0 Network: 1	50000
P2	Report: 1 Data: 0 Algorithm: 0	35000

Step 6: Then, we pick the next expert with the minimum communication cost with the core expert to fulfill the remaining required skills for the set of projects. Expert B is chosen as it has the least communication cost with the core expert. Expert B can cover two skills (P1-Network, P2-Report) from the required skillset for the projects.

Budget remaining: 25000 – Cost of an expert B = 25000 – 7000 = 18000

Group of experts: {E, A, F, B}

Table 9: Step 6 - Remaining projects' skills

Projects	**Required Skills: Required number of experts**	**Profit**
P1	Java: 0 Network: 0	50000
P2	Report: 0 Data: 0 Algorithm: 0	35000

After step 6, all the requirements for the given projects are fulfilled as seen in Table 9. Hence, our final group of experts for the given two projects will be,

Final Group of experts: {E, A, F, B}

We were able to hire above mentioned group of experts for the given set of projects, within the given budget of 50,000 while keeping the communication cost between the experts minimal. This is one example of solution to cluster hire problem in social networks using MWSCAN. In the next chapter, we will discuss the Implementation details with results and analysis.

CHAPTER 4
RESULT AND ANALYSIS

In this chapter, we are going to evaluate the performance of our proposed approach to solve the problem of Cluster Hire in social networks.

4.1 Experimental Setup

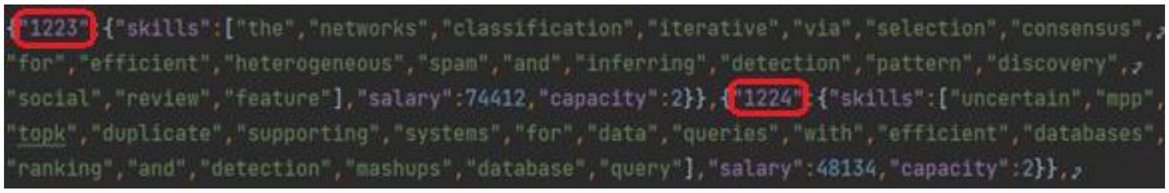
All the experiments were conducted on a computer with device specifications of Intel Core i7-6700 CPU @ 3.40 GHz, on a 64-bit Windows operating system with 8 GB RAM. We implemented our algorithm in JDK 14.0.1 (windows-64 bit) using IntelliJ IDEA 2020.1.3.

For the purpose of testing our algorithm on a real social network, we are using the DBLP dataset, which is one of the experts' network also used in [21] and [30]. DBLP is a computer science bibliography website that contains a dataset of over 5.4 million publications, journals, and conference papers on computer science, according to dblp.org. In our setup, we model the DBLP dataset as a social network graph, where nodes represent authors/experts. When two experts publish any paper together, they will have the connection/edge between them. The edge has a weight associated with it, which is called communication cost. If two experts are not directly connected, then we use weight of the shortest path between them as the communication cost. The dataset contains information about 50K experts.

The experts' dataset is stored in JSON format, where each expert has four attributes, namely expert ID, skills, cost/salary, and capacity. The expert ID is a unique identifier for each expert. Each expert has a set of skills, which they can offer. The skills of experts are extracted from the titles of the research articles/publications of experts on the DBLP network. The hiring cost of an expert is the salary or monetary value assigned to the expert. The hiring cost of an expert is randomly set up to 100,000 in our dataset. Each expert also has a working capacity, which indicates the number of projects, an expert can work on simultaneously. The working capacity of an expert is randomly set between 1 to 5. We use the different number of projects to generate results for the experiments like 3, 5, 7, 10, and 14 number of projects. For each project, we assign 3 to 7 skills that must be fulfilled to complete a project. We must assign one expert to each required skill of the project, which means that number of experts assigned to the project will be equal to the number of skills required for the project. One expert cannot cover multiple skills in the same project, i.e., each expert will fulfill unique skill in the given project. Each project has an assigned value of profit, which is generated on the completion of the project. We set the priority for the projects based on their profit value. In some cases, we cannot cover all the projects in a given set of projects due to budget constraints; then, we hire the experts for the higher profit projects first and then for projects with lower profit value. In our experiments, we also use different values for the budget to analyze the results returned by our algorithm.

The proposed algorithm has several dataset files as an input. Below is the list of input dataset files to the MWSCAN algorithm for the Cluster Hire problem in social networks,

1. A file containing information about experts, where we have a dataset consisting of 50K experts/nodes. In Figure 11, we have information about two experts, and you can identify the experts by their expert IDs, which are 1223 and 1224, highlighted in red color. For each expert ID, we have three attributes, which are skills, salary, and capacity. These information about experts is shown in Table 10 as well. The skills of an expert are displayed as an array of skills for each expert ID. Then, we have values of salary and capacity for each expert ID.



```
{"1223":{"skills":["the","networks","classification","iterative","via","selection","consensus",
"for","efficient","heterogeneous","spam","and","inferring","detection","pattern","discovery",
"social","review","feature"],"salary":74412,"capacity":2}},{"1224":{"skills":["uncertain","mpp",
"topk","duplicate","supporting","systems","for","data","queries","with","efficient","databases",
"ranking","and","detection","mashups","database","query"],"salary":48134,"capacity":2}},
```

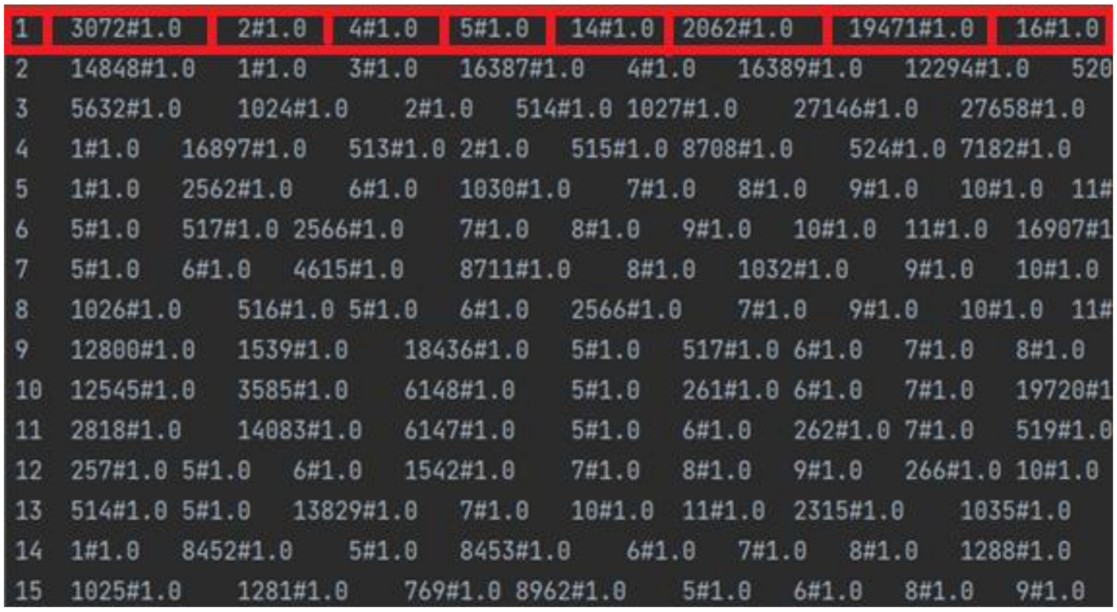
Figure 11: Experts JSON file (a)

Table 10: Experts JSON file (b)

Expert ID	**Skills**	**Salary/ Cost**	**Capacity**
1223	"the","networks","classification","iterative","via", "selection","consensus","for","efficient","heterogeneous","spam","and","inferring","detection","pattern","discovery","social","review","feature"	74412	2

1224	"uncertain","mpp","topk","duplicate","supporting","systems","for","data","queries","with","efficient","databases","ranking","and","detection","mashups","database","query"	48134	2

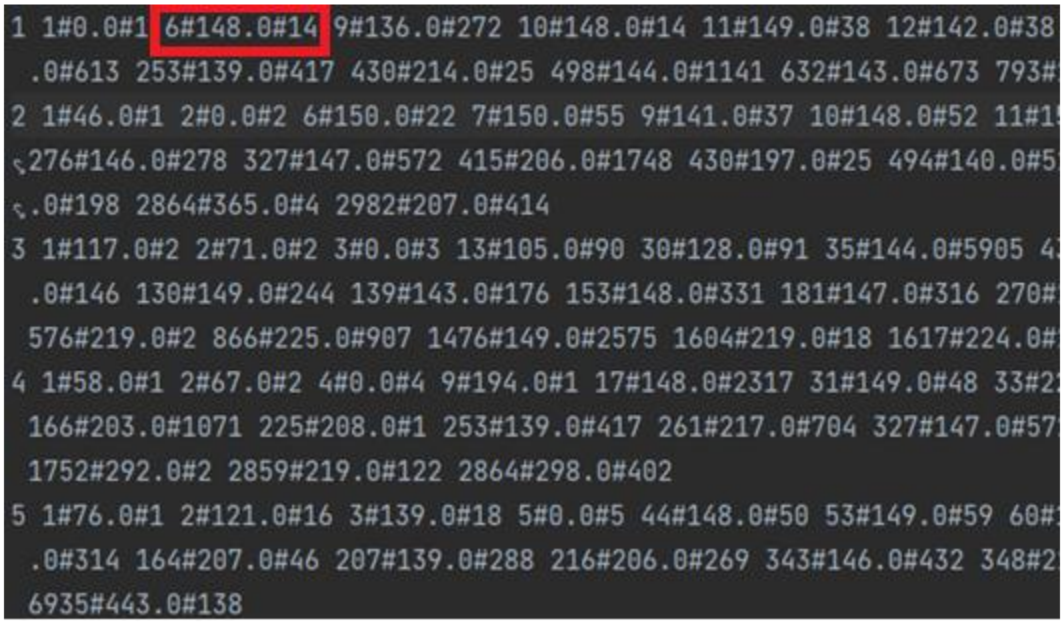
2. A file containing information about connections/edges between experts in a social network of experts. Social network of experts is where nodes represent experts, and experts are connected with edges, which are weighted. If two experts are connected in a social network, it means they have a connection and they have collaborated in the past. If there exists a connection between two experts, then it assigns a value of 1.0 between experts. The value of 1.0 does not represent edge weight between experts, but it just means that they have a connection and they have collaborated in the past. In Figure 12, if we look at the first line which is highlighted in red, it displays all the connections expert ID '1' has with other experts in a social network. We have highlighted each connection separately in red color for expert ID '1'. For example, first element in the highlighted section in Figure 12, '3072#1.0' means that expert ID '3072' is assigned value of 1.0, which means there exists a connection between expert IDs '1' and '3072'. So, it has a list of expert IDs like 3072, 2, 4, 5, 14, 2062, 19471 and 16 with the value of 1.0, which means these experts have connection with expert ID '1'. This way, the file stores information about all the experts and their connections in the social network of experts. This file helps us to count the number of connections for any expert in the dataset.



```
1   3072#1.0     2#1.0   4#1.0   5#1.0   14#1.0  2062#1.0     19471#1.0   16#1.0
2   14848#1.0    1#1.0   3#1.0   16387#1.0   4#1.0   16389#1.0   12294#1.0   520
3   5632#1.0     1024#1.0    2#1.0   514#1.0 1027#1.0    27146#1.0   27658#1.0
4   1#1.0   16897#1.0   513#1.0 2#1.0   515#1.0 8708#1.0    524#1.0 7182#1.0
5   1#1.0   2562#1.0    6#1.0   1030#1.0    7#1.0   8#1.0   9#1.0   10#1.0  11#
6   5#1.0   517#1.0 2566#1.0    7#1.0   8#1.0   9#1.0   10#1.0  11#1.0  16907#1
7   5#1.0   6#1.0   4615#1.0    8711#1.0    8#1.0   1032#1.0    9#1.0   10#1.0
8   1026#1.0    516#1.0 5#1.0   6#1.0   2566#1.0    7#1.0   9#1.0   10#1.0  11#
9   12800#1.0   1539#1.0    18436#1.0   5#1.0   517#1.0 6#1.0   7#1.0   8#1.0
10  12545#1.0   3585#1.0    6148#1.0    5#1.0   261#1.0 6#1.0   7#1.0   19720#1
11  2818#1.0    14083#1.0   6147#1.0    5#1.0   6#1.0   262#1.0 7#1.0   519#1.0
12  257#1.0 5#1.0   6#1.0   1542#1.0    7#1.0   8#1.0   9#1.0   266#1.0 10#1.0
13  514#1.0 5#1.0   13829#1.0   7#1.0   10#1.0  11#1.0  2315#1.0    1035#1.0
14  1#1.0   8452#1.0    5#1.0   8453#1.0    6#1.0   7#1.0   8#1.0   1288#1.0
15  1025#1.0    1281#1.0    769#1.0 8962#1.0    5#1.0   6#1.0   8#1.0   9#1.0
```

Figure 12: Connections in social network of experts file

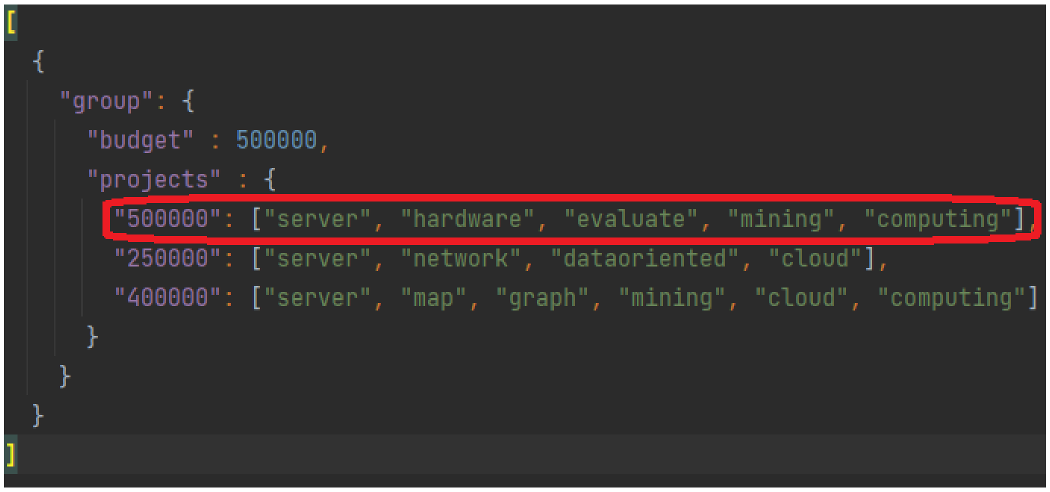
3. A file containing information about the communication cost value between experts in the social network of experts. As discussed earlier, in a network of experts, experts are connected to each other via weighted edge. The communication cost can be defined as a weight/distance of an edge between the experts, and the value of communication cost is based on the past collaboration between experts. If two experts are not directly connected, then we use the value of the shortest path between them as the communication cost. The lower value of communication cost signifies more collaborative experts. This file is generated using a 2-hop cover approach proposed as discussed in [32] and [33]. The 2-hop cover helps with calculating the shortest path distance between nodes in large networks; otherwise, it can be very time-consuming. In Figure 13, we are displaying communication cost values between various expert IDs in a social network of experts. For example, if we pick an element highlighted in red color from Figure 13, which is "6#148.0#14", it means that the value of communication cost between expert ID '6' and '14' is 148.0. In this format, the communication cost information is stored between all experts in social network of experts.



```
1 1#0.0#1 6#148.0#14 9#136.0#272 10#148.0#14 11#149.0#38 12#142.0#38
 .0#613 253#139.0#417 430#214.0#25 498#144.0#1141 632#143.0#673 793#
2 1#46.0#1 2#0.0#2 6#150.0#22 7#150.0#55 9#141.0#37 10#148.0#52 11#1
 276#146.0#278 327#147.0#572 415#206.0#1748 430#197.0#25 494#140.0#5
 .0#198 2864#365.0#4 2982#207.0#414
3 1#117.0#2 2#71.0#2 3#0.0#3 13#105.0#90 30#128.0#91 35#144.0#5905 4
 .0#146 130#149.0#244 139#143.0#176 153#148.0#331 181#147.0#316 270#
 576#219.0#2 866#225.0#907 1476#149.0#2575 1604#219.0#18 1617#224.0#
4 1#58.0#1 2#67.0#2 4#0.0#4 9#194.0#1 17#148.0#2317 31#149.0#48 33#2
 166#203.0#1071 225#208.0#1 253#139.0#417 261#217.0#704 327#147.0#57
 1752#292.0#2 2859#219.0#122 2864#298.0#402
5 1#76.0#1 2#121.0#16 3#139.0#18 5#0.0#5 44#148.0#50 53#149.0#59 60#
 .0#314 164#207.0#46 207#139.0#288 216#206.0#269 343#146.0#432 348#2
 6935#443.0#138
```

Figure 13: Communication cost file

4. A file with information about projects requirements such as each project's required skills, profit of each project, and budget assigned for the given set of projects. In Figure 14 below, we have a group of three projects, for which we have to hire a group of experts. The budget assigned for hiring for the given group of projects is 500,000. Each project has assigned profit and set of required skills, which is stored in key and value pair. For example, if we look at the first project in Figure 14, which is highlighted in red color. The profit of this projects is 500,000, and the set of required skills is {"server", "hardware", "evaluate", "mining", "computing"}. Information about the other two projects is stored in the same format. Please note that we use the profit value of each project just to set priority between projects. For example, if we cannot complete all the projects in the given budget, we prioritize the projects with higher profits.



```
[
  {
    "group": {
      "budget" : 500000,
      "projects" : {
        "500000": ["server", "hardware", "evaluate", "mining", "computing"],
        "250000": ["server", "network", "dataoriented", "cloud"],
        "400000": ["server", "map", "graph", "mining", "cloud", "computing"]
      }
    }
  }
]
```

Figure 14: Project requirements file

The primary objective of our thesis is to study and analyze the impact of the MWSCAN algorithm when applied to the problem of Cluster Hire in social networks, which is an NP-hard problem [23, 31]. We evaluate the performance of MWSCAN algorithm in terms of number of projects completed, Profit generated from the completed projects and the runtime of the algorithm for different values of budget and number of projects. Furthermore, we compare our results with the previous greedy algorithm proposed by Sagarika et al. [31] to solve the cluster Hire problem in social networks.

4.2 Results

In this section, we will discuss the results obtained by performing various experiments for the given dataset using the proposed algorithm MWSCAN. The dataset of 50K experts and the social network of experts will remain the same for all the experiments, and we vary the number of projects in a group for our experiments. We will be performing experiments on different number of projects in a group, as shown in Table 11 below. For each experiment, we will be analyzing an output based on the number of projects completed, the runtime of the algorithm and also making sure that expenses of hiring experts are within the given budget.

Table 11: Cases for experiments

Experiment Number	**Number of projects**	**Budget**
1	3	500k

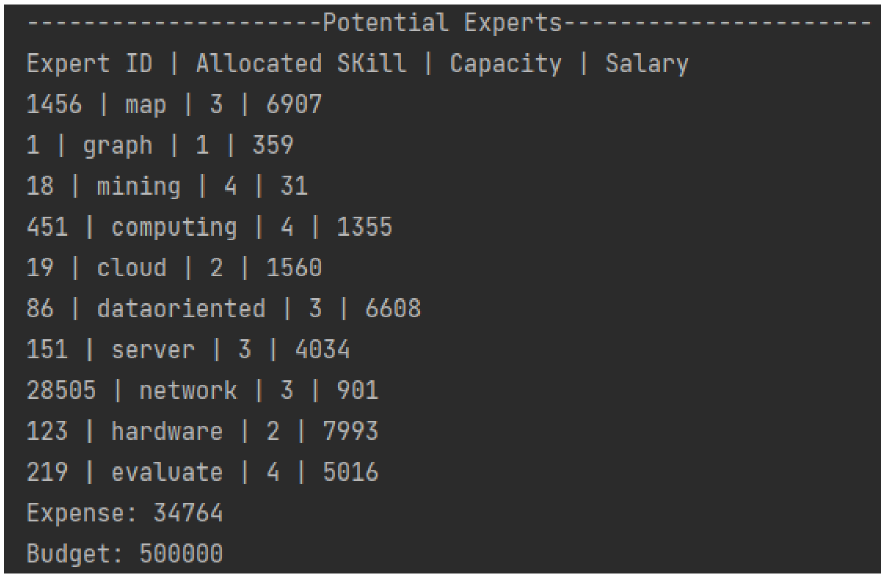
2	5	500k
3	7	500k
4	10	500k
5	14	500k

In experiment 1, we have a set of three projects for which we are looking to hire a group of experts to fulfill the skills required for the projects. We have set the value of the budget to 500,000 for the given set of three projects. Table 12 below shows the projects requirements information for the experiment 1. We have all the information about the projects like their required set of skills, budget assigned for a group of projects, and profit of projects.

Table 12: Projects requirements for experiment 1

Projects	**Required skills**	**Profit**	**Budget**
Project 1	server, hardware, evaluate, mining, computing	500k	500k
Project 2	server, network, dataoriented, cloud	250k	
Project 3	server, map, graph, mining, cloud, computing	400k	

For the given group of projects, we are able to hire a group of experts with the minimum communication cost to complete all three projects. The cost of hired group of experts is within the given budget and we are hiring the experts who are collaborative i.e., they have minimum communication cost between them. As shown in Figure 15, the expense of hiring the group of experts is just 34,764 which is within the given budget, therefore we were able to complete all three projects from the given set of projects. The runtime for the given experiment is 59 seconds. The output for experiment 1 is showed in the Figure 15, where we can see list of hired experts along with their information like their expert ID, allocated skill, capacity, and cost/salary. In Table 13 below, we display the assigned expert for each required skill for the given three projects.



```
---------------------Potential Experts----------------------
Expert ID | Allocated SKill | Capacity | Salary
1456 | map | 3 | 6907
1 | graph | 1 | 359
18 | mining | 4 | 31
451 | computing | 4 | 1355
19 | cloud | 2 | 1560
86 | dataoriented | 3 | 6608
151 | server | 3 | 4034
28505 | network | 3 | 901
123 | hardware | 2 | 7993
219 | evaluate | 4 | 5016
Expense: 34764
Budget: 500000
```

Figure 15: Result for experiment 1

Table 13: Projects assignments for experiment 1

Projects	**Required skills**	**Assigned expert ID**	**Profit**
Project 1	server	151	500k
	hardware	123	
	evaluate	219	
	mining	18	
	computing	451	
Project 2	server	151	250k
	network	28505	
	dataoriented	86	
	hardware	123	
Project 3	server	151	400k
	map	1456	
	graph	1	
	mining	18	

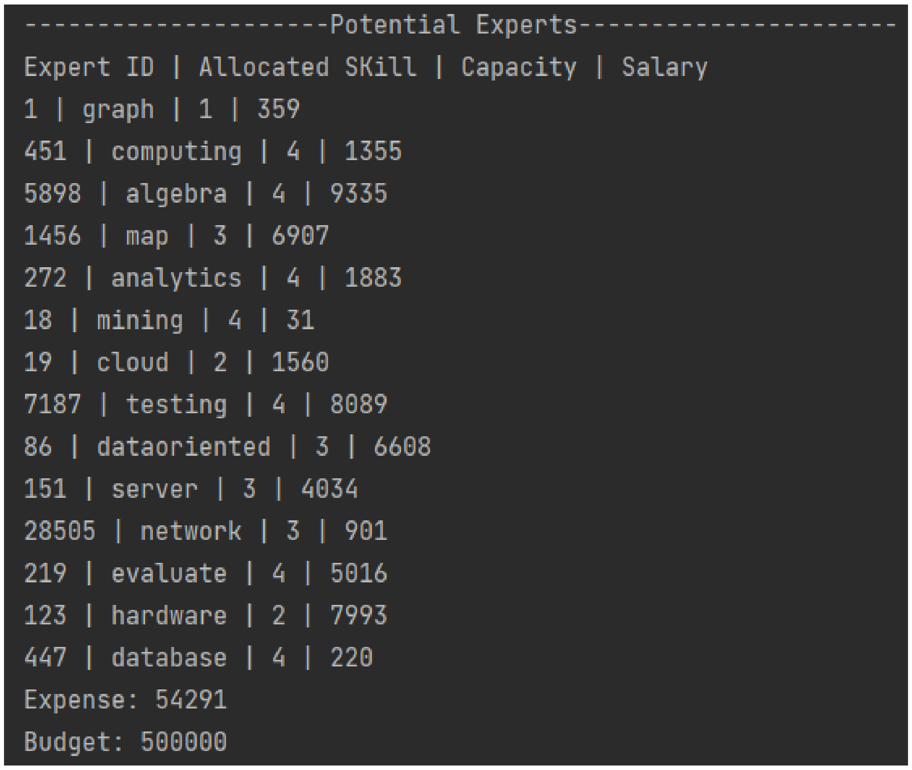
	cloud	19	
	computing	451	

For the experiment 2, we increase the number of projects to five projects, but we keep the value of budget constant. Table 14 below represents the projects requirements information for the second experiment. We have a greater number of projects here and more skills to cover while hiring the experts than experiment 1.

Table 14: Projects requirements for experiment 2

Projects	**Required skills**	**Profit**	**Budget**
Project 1	server, hardware, evaluate, mining, computing	500k	500k
Project 2	server, network, dataoriented, cloud	250k	
Project 3	server, map, graph, mining, cloud, computing	400k	
Project 4	analytics, database, computing, testing	275k	
Project 5	algebra, computing, map	200k	

Figure 16 below depicts the output for the experiment 2. It lists the group of experts who fulfills the skills required for the given five projects. Again, these hired experts are within the given budget and they are hired in a way that the group of experts are collaborative with minimum communication cost between the hired experts. The expense of the hiring of experts is 54291, which is within the budget of 500,000. The runtime for this experiment is 14 minutes 13 seconds, which is considerably higher than the previous case. We observe that a greater number of experts are hired, and they fulfill all the required skills with reasonable cost. Also, we display the individual expert assignment for each skill in the given group of projects in Table 15 below.



```
---------------------Potential Experts----------------------
Expert ID | Allocated SKill | Capacity | Salary
1 | graph | 1 | 359
451 | computing | 4 | 1355
5898 | algebra | 4 | 9335
1456 | map | 3 | 6907
272 | analytics | 4 | 1883
18 | mining | 4 | 31
19 | cloud | 2 | 1560
7187 | testing | 4 | 8089
86 | dataoriented | 3 | 6608
151 | server | 3 | 4034
28505 | network | 3 | 901
219 | evaluate | 4 | 5016
123 | hardware | 2 | 7993
447 | database | 4 | 220
Expense: 54291
Budget: 500000
```

Figure 16: Result for experiment 2

Table 15: Projects assignments for experiment 2

Projects	**Required skills**	**Assigned expert ID**	**Profit**
Project 1	server	151	500k
	hardware	123	
	evaluate	219	
	mining	18	
	computing	451	
Project 2	server	151	250k
	network	28505	
	dataoriented	86	
	hardware	123	
Project 3	server	151	400k

	map	1456	
	graph	1	
	mining	18	
	cloud	19	
	computing	451	
Project 4	analytics	272	275k
	database	447	
	computing	451	
	testing	7187	
Project 5	algebra	5898	200k
	computing	451	
	map	1456	

In experiment 3, the number of projects is increased to group of seven projects. As shown in Table 16, budget is set to 500,000. We have set of required skills and profit value for each project in a group as shown in Table 16.

Table 16: Projects requirements for experiment 3

Projects	**Required skills**	**Profit**	**Budget**
Project 1	server, hardware, evaluate, mining, computing	500k	500k
Project 2	server, network, dataoriented, cloud	250k	
Project 3	server, map, graph, mining, cloud, computing	400k	
Project 4	analytics, database, computing, testing	275k	
Project 5	algebra, computing, map	200k	
Project 6	cloud, infrastructure, design, database	300k	
Project 7	algorithm, testing, design	350k	

After running our algorithm for experiment 3, we were able to hire experts for all seven projects within the given budget. The group of experts are hired while keeping the communication cost minimum between them. As shown in Figure 17, the expense of hiring experts is 56172, which is within the given budget. The runtime for the given experiment was 18 minutes 28 seconds. In Table 17, we show the individual expert assignment for each required skill in the given group of projects. We show the expert ID assigned for each required skill from our output.

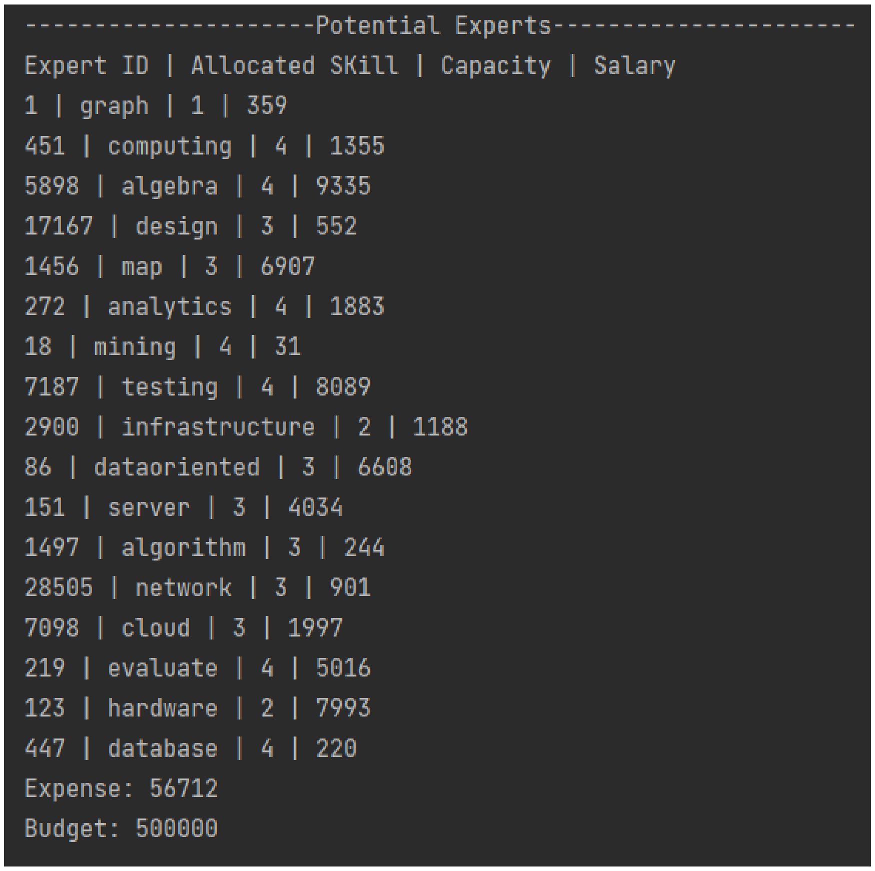

```
---------------------Potential Experts----------------------
Expert ID | Allocated SKill | Capacity | Salary
1 | graph | 1 | 359
451 | computing | 4 | 1355
5898 | algebra | 4 | 9335
17167 | design | 3 | 552
1456 | map | 3 | 6907
272 | analytics | 4 | 1883
18 | mining | 4 | 31
7187 | testing | 4 | 8089
2900 | infrastructure | 2 | 1188
86 | dataoriented | 3 | 6608
151 | server | 3 | 4034
1497 | algorithm | 3 | 244
28505 | network | 3 | 901
7098 | cloud | 3 | 1997
219 | evaluate | 4 | 5016
123 | hardware | 2 | 7993
447 | database | 4 | 220
Expense: 56712
Budget: 500000
```

Figure 17: Result for experiment 3

Table 17: Project assignments for experiment 3

Projects	**Required skills**	**Assigned expert ID**	**Profit**
Project 1	server	151	500k
	hardware	123	
	evaluate	219	

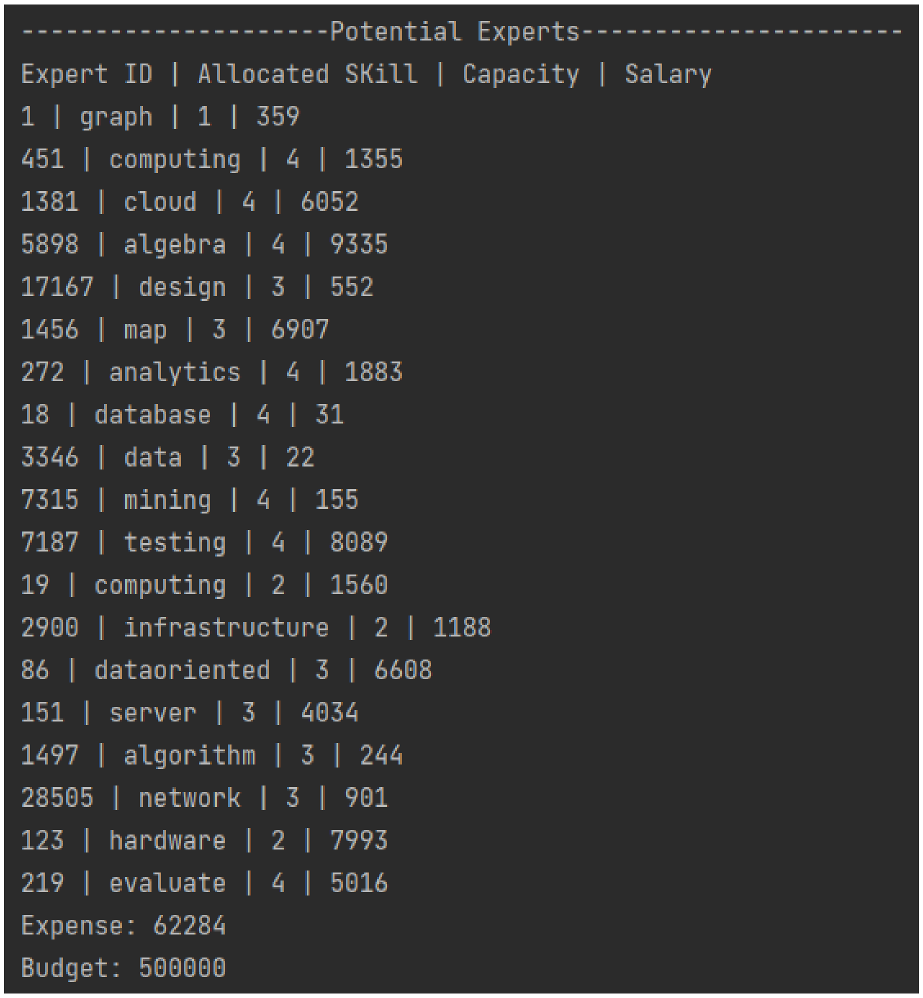
	mining	18	
	computing	451	
Project 2	server	151	250k
	network	28505	
	dataoriented	86	
	hardware	123	
Project 3	server	151	400k
	map	1456	
	graph	1	
	mining	18	
	cloud	7098	
	computing	451	
Project 4	analytics	272	275k
	database	447	
	computing	451	
	testing	7187	
Project 5	algebra	5898	200k
	computing	451	
	map	1456	
Project 6	cloud	7098	300k
	infrastructure	2900	
	design	17167	
	database	447	
Project 7	algorithm	1497	350k
	testing	7187	
	design	17167	

In experiment 4, we are hiring a group of experts for ten projects. We are keeping the value of budget constant to 500,000 as shown in Table 18. The goal is to hire a group of experts for all ten projects with minimum communication cost within the given budget. We will also keep track of the execution time of our algorithm.

Table 18: Projects requirements for experiment 4

Projects	**Required skills**	**Profit**	**Budget**
Project 1	server, hardware, evaluate, mining, computing	500k	500k
Project 2	server, network, dataoriented, cloud	250k	
Project 3	server, map, graph, mining, cloud, computing	400k	
Project 4	analytics, database, computing, testing	275k	
Project 5	algebra, computing, map	200k	
Project 6	cloud, infrastructure, design, database	300k	
Project 7	algorithm, testing, design	350k	
Project 8	database, mining, cloud	375k	
Project 9	algorithm, computing, testing, data	180k	
Project 10	cloud, database, network, algorithm	420k	

After running the MWSCAN algorithm for the given ten projects, we were able to successfully hire a group of experts for all 10 projects within the given budget. Figure 18 shows the output for experiment 4 with information of all the hired experts and the total cost for hiring the experts. The expense of hiring experts is 62284, which is within the given budget. The runtime for this particular experiment is 36 minutes as we increase the number of projects to ten. But given the NP-hard nature of the problem and large dataset of 50k nodes, the runtime is expected to increase with increase in number projects. In Table 19, we assign expert IDs to each required skill in the given group of projects along with their profit value.



```
---------------------Potential Experts----------------------
Expert ID | Allocated SKill | Capacity | Salary
1 | graph | 1 | 359
451 | computing | 4 | 1355
1381 | cloud | 4 | 6052
5898 | algebra | 4 | 9335
17167 | design | 3 | 552
1456 | map | 3 | 6907
272 | analytics | 4 | 1883
18 | database | 4 | 31
3346 | data | 3 | 22
7315 | mining | 4 | 155
7187 | testing | 4 | 8089
19 | computing | 2 | 1560
2900 | infrastructure | 2 | 1188
86 | dataoriented | 3 | 6608
151 | server | 3 | 4034
1497 | algorithm | 3 | 244
28505 | network | 3 | 901
123 | hardware | 2 | 7993
219 | evaluate | 4 | 5016
Expense: 62284
Budget: 500000
```

Figure 18: Result for experiment 4

Table 19: Projects assignments for experiment 4

Projects	**Required skills**	**Assigned expert ID**	**Profit**
Project 1	server	151	500k
	hardware	123	
	evaluate	219	
	mining	18	
	computing	451	
Project 2	server	151	250k

	network	28505	
	dataoriented	86	
	hardware	123	
Project 3	server	151	400k
	map	1456	
	graph	1	
	mining	18	
	cloud	19	
	computing	451	
Project 4	analytics	272	275k
	database	447	
	computing	451	
	testing	7187	
Project 5	algebra	5898	200k
	computing	451	
	map	1456	
Project 6	cloud	7098	300k
	infrastructure	2900	
	design	17167	
	database	447	
Project 7	algorithm	1497	350k
	testing	7187	
	design	17167	
Project 8	cloud	7098	375k

	infrastructure	2900	
	design	17167	
	database	447	
Project 9	algorithm	1497	180k
	testing	7187	
	design	17167	
Project 10	cloud	7098	420k
	infrastructure	2900	
	design	17167	
	database	447	

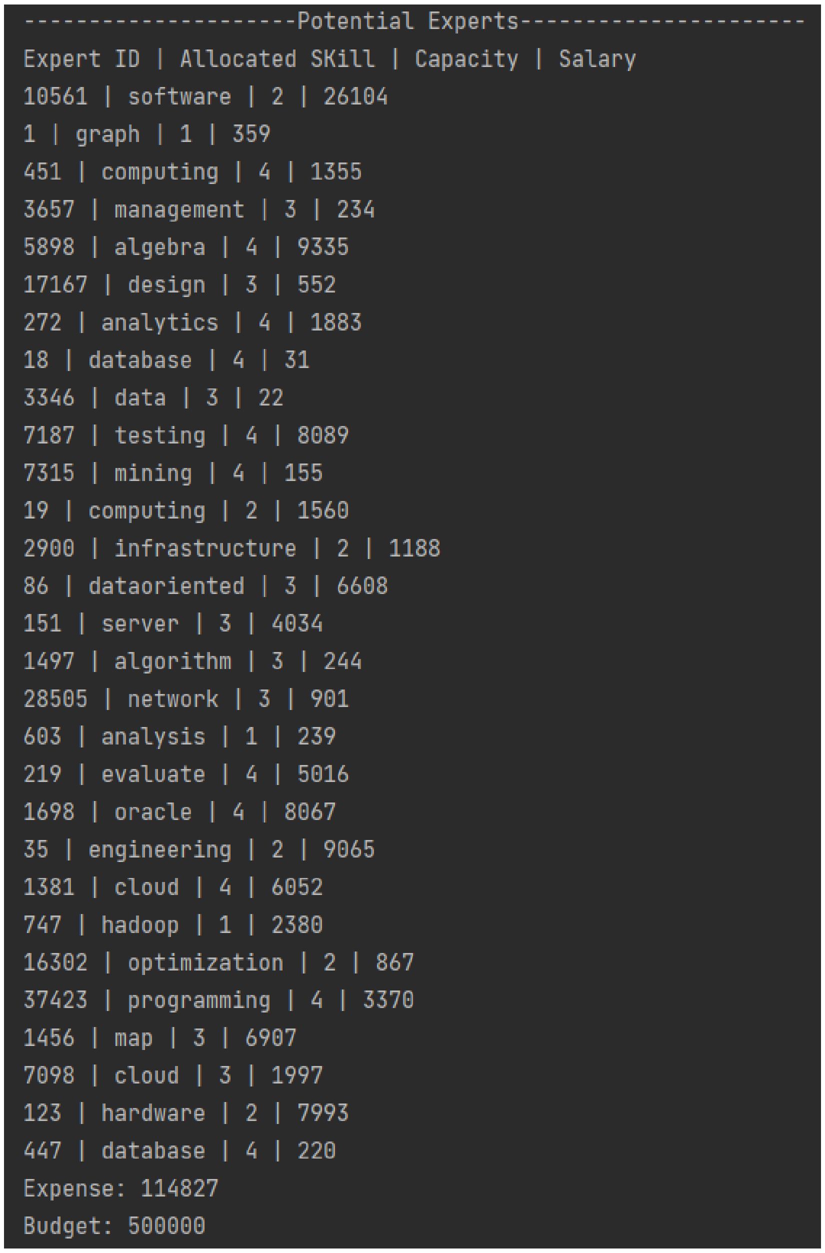
In experiment 5, we are increasing the number of projects to fourteen to see how our algorithm performs for a large number of projects. As shown in Table 20, we have a group of fourteen projects with the budget assigned for a group of projects, profit of each project and set of required skills for each project.

Table 20: Projects requirements for experiment 5

Projects	**Required skills**	**Profit**	**Budget**
Project 1	server, hardware, evaluate, mining, computing	500k	500k
Project 2	server, network, dataoriented, cloud	250k	
Project 3	server, map, graph, mining, cloud, computing	400k	
Project 4	analytics, database, computing, testing	275k	
Project 5	algebra, computing, map	200k	
Project 6	cloud, infrastructure, design, database	300k	
Project 7	algorithm, testing, design	350k	
Project 8	database, mining, cloud	375k	
Project 9	algorithm, computing, testing, data	180k	
Project 10	cloud, database, network, algorithm	420k	

Project 11	computing, optimization, analysis, management	450k	
Project 12	optimization, software, testing	390k	
Project 13	oracle, hadoop, database, network	475k	
Project 14	software, programming, oracle, engineering	225k	

After running the MWSCAN algorithm for the experiment 5, we are still able to hire a group of experts for the given group of fourteen projects within the given budget and with minimum communication cost. As shown in Figure 19, the expense of hiring the experts is 114,827, which is within the assigned budget of 500,000. However, the runtime for experiment 5 was 59 minutes, which was considerably high. The high runtime is expected as the number of projects increase and also given the large network of experts. In Table 21, we show the assigned expert IDs to each required skill in the given set of projects.



```
---------------------Potential Experts----------------------
Expert ID | Allocated SKill | Capacity | Salary
10561 | software | 2 | 26104
1 | graph | 1 | 359
451 | computing | 4 | 1355
3657 | management | 3 | 234
5898 | algebra | 4 | 9335
17167 | design | 3 | 552
272 | analytics | 4 | 1883
18 | database | 4 | 31
3346 | data | 3 | 22
7187 | testing | 4 | 8089
7315 | mining | 4 | 155
19 | computing | 2 | 1560
2900 | infrastructure | 2 | 1188
86 | dataoriented | 3 | 6608
151 | server | 3 | 4034
1497 | algorithm | 3 | 244
28505 | network | 3 | 901
603 | analysis | 1 | 239
219 | evaluate | 4 | 5016
1698 | oracle | 4 | 8067
35 | engineering | 2 | 9065
1381 | cloud | 4 | 6052
747 | hadoop | 1 | 2380
16302 | optimization | 2 | 867
37423 | programming | 4 | 3370
1456 | map | 3 | 6907
7098 | cloud | 3 | 1997
123 | hardware | 2 | 7993
447 | database | 4 | 220
Expense: 114827
Budget: 500000
```

Figure 19: Result for experiment 5

Table 21: Project assignments for experiment 5

Projects	**Required skills**	**Assigned expert ID**	**Profit**
Project 1	server	151	500k
	hardware	123	
	evaluate	219	

	mining	18	
	computing	451	
Project 2	server	151	250k
	network	28505	
	dataoriented	86	
	hardware	123	
Project 3	server	151	400k
	map	1456	
	graph	1	
	mining	18	
	cloud	19	
	computing	451	
Project 4	analytics	272	275k
	database	447	
	computing	451	
	testing	7187	
Project 5	algebra	5898	200k
	computing	451	
	map	1456	
Project 6	cloud: 7098	7098	300k
	infrastructure	2900	
	design	17167	
	database	447	
Project 7	algorithm	1497	350k
	testing	7187	
	design	17167	

Project 8	cloud	7098	375k
	infrastructure	2900	
	design	17167	
	database	447	
Project 9	algorithm	1497	180k
	testing	7187	
	design	17167	
Project 10	cloud	7098	420k
	infrastructure	2900	
	design	17167	
	database	447	
Project 11	computing	19	450k
	optimization	16302	
	analysis	603	
	management	3657	
Project 12	optimization	16302	390k
	software	10561	
	testing	7187	
Project 13	oracle	1698	475k
	hadoop	787	
	database	18	
	network	28505	
Project 14	software	10561	225k
	programming	37423	
	oracle	1698	
	engineering	35	

For all five experiments, the proposed algorithm successfully provided a solution to the cluster hire problem in social networks. The results were produced meeting all the constraints of the problem like hiring the experts within the assigned budget and their working capacity, completing all the skills of the given projects, and minimizing the communication cost between the hired group of experts. Now, we will compare our results with the past approaches used to solve the cluster hire problem in social networks.

4.3 Comparison with Greedy algorithm

The aim of our thesis is to study and analyze the effect of the MWSCAN algorithm on solving the Cluster Hire problem in social networks. In the past, greedy algorithms and ILP approach were proposed to solve the problem of Cluster Hire in social networks. While greedy algorithms produced faster results for the Cluster Hire problem in social networks, it was not an optimal or near-optimal solution to the problem. As greedy algorithms make locally optimal choices at each stage, it fails to generate global optimal output. We implemented MWSCAN algorithm to solve the Cluster Hire problem in social networks. We performed various experiments for the MWSCAN algorithm on a large network of 50K experts and compared our results with the greedy algorithm implemented by Sagarika et al. [31].

4.3.1 Number of projects completed vs. Budget

To measure the effectiveness of the MWSCAN algorithm, we compare the number of projects completed within the given budget for both the MWSCAN and the greedy algorithm. We performed experiments for k number of projects, where $k = \{10,14\}$. In this setup, the database of 50K experts and the graph of experts remain the same, and we just change the value of the budget assigned to hire experts. Then, we analyze the number of projects the algorithms can complete within the given budget. Figure 20 represents the graph for the number of projects completed vs Budget for 10 number of projects for both MWSCAN and the greedy algorithm. As shown in the graph, the MWSCAN algorithm was able to complete a greater number of projects for the given budget compared to the greedy algorithm. It means that MWSCAN can prove more cost-efficient to organizations as they can complete a greater number of projects in a limited budget than the solution provided by greedy algorithms.

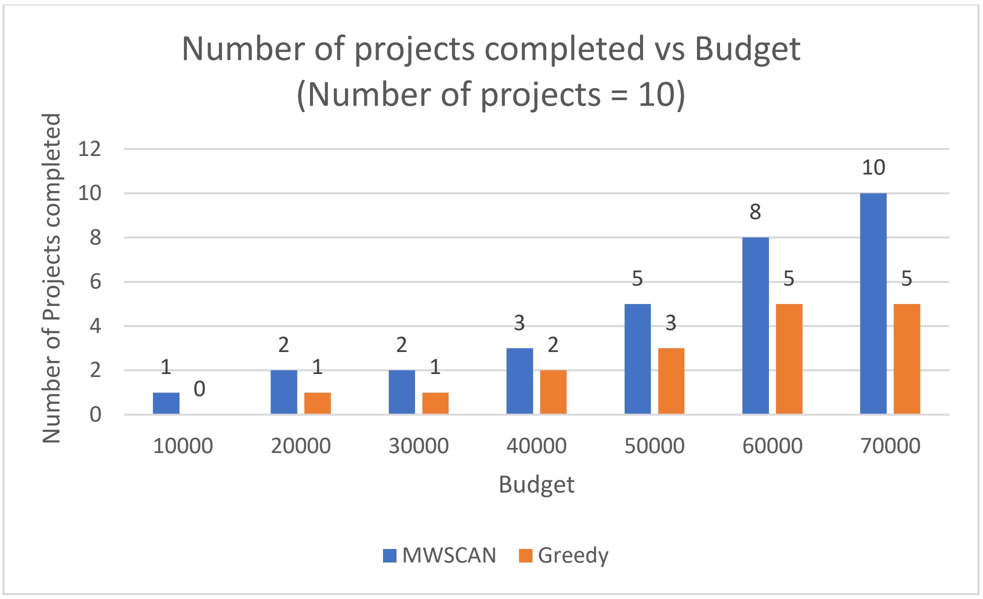


Figure 20: Number of projects completed vs Budget (Number of projects = 10)

For the second case of the experiment, we change the number of projects in a group to 14. We analyze the number of projects completed for varying values of budget. In Figure 21, we are showing the results for the number of projects completed vs budget for both MWSCAN and greedy algorithm. Once again, MWSCAN completes more number of projects compared to the greedy algorithm for the given budget. The results generated by MWSCAN are cost-effective and profitable for the organizations. As MWSCAN completes more projects within the limited budget, it generates more profit for the company. We will discuss the profit comparison between the two approaches in the next section.

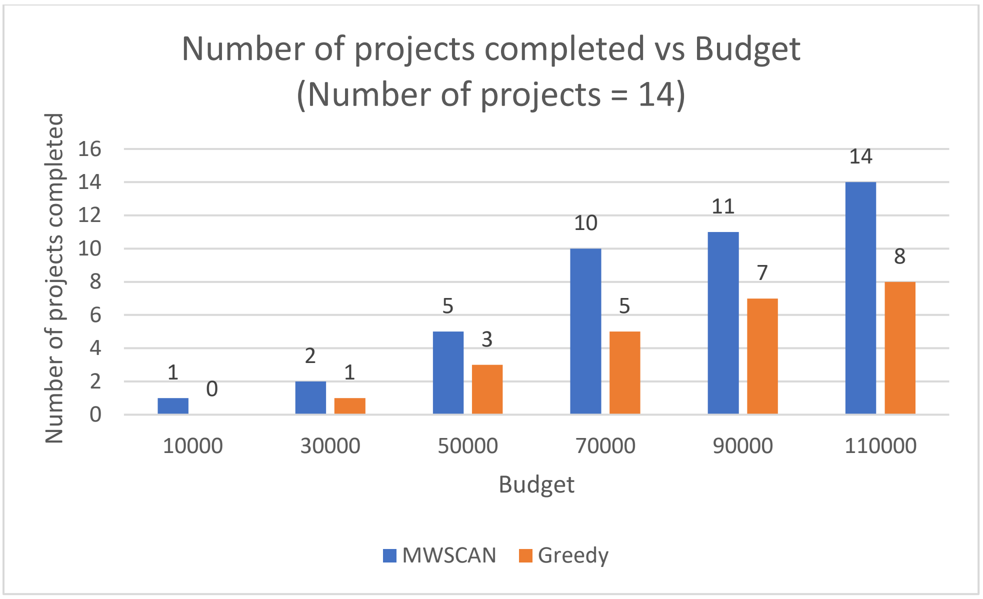


Figure 21: Number of projects completed vs Budget (Number of projects = 14)

4.3.2 Total Profit vs. Budget

In this section, we are comparing the profit generated by the MWSCAN algorithm vs the greedy algorithm for the varying value of the budget. For these experiments, we keep the experts and graph of experts dataset the same. We change the number of projects and perform the experiments for k number of projects, where $k = \{10,14\}$. Figure 22 displays a graph of the total profit generated vs budget for 10 number of projects. The graph shows the comparison between the MWSCAN and the greedy algorithm. As can be seen from the graph, the total profit generated for the MWSCAN algorithm is higher than the greedy algorithm.

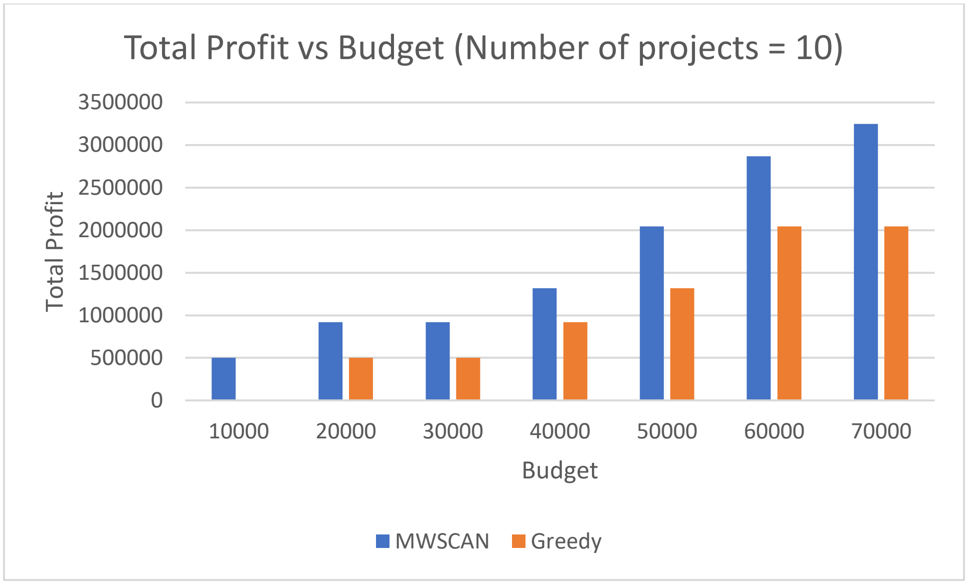


Figure 22: Total Profit vs Budget (Number of projects = 10)

For the second case, we have a group of 14 projects, and we analyze the total profit generated vs budget for both the MWSCAN and the greedy algorithm. Figure 23 below shows the profit comparison results between the two algorithms. It is clear from Figure 23 that the MWSCAN algorithm generates more total profit compared to the greedy algorithm for various values of the budget. From our experiments, we can say that the MWSCAN algorithm is more profitable for organizations compared to the greedy algorithm.

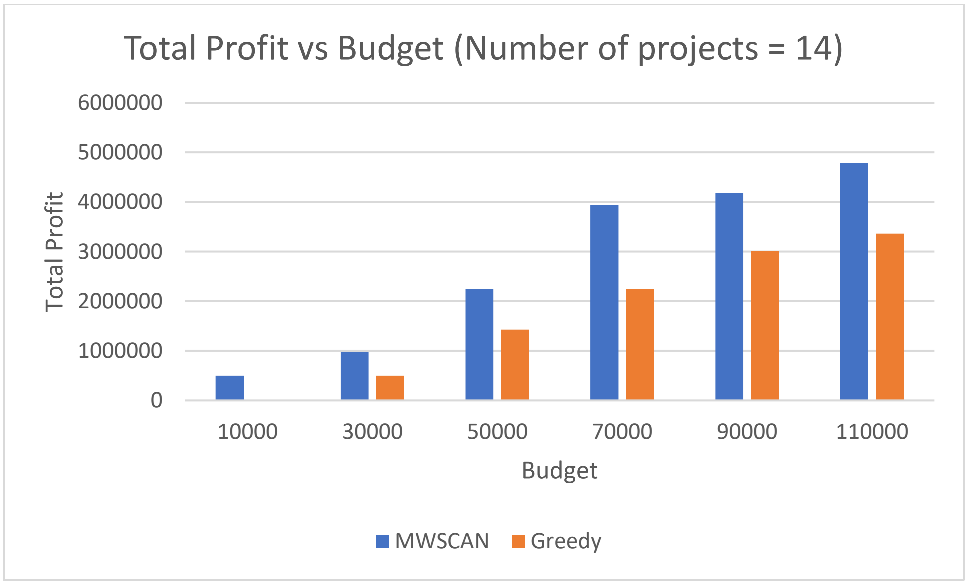
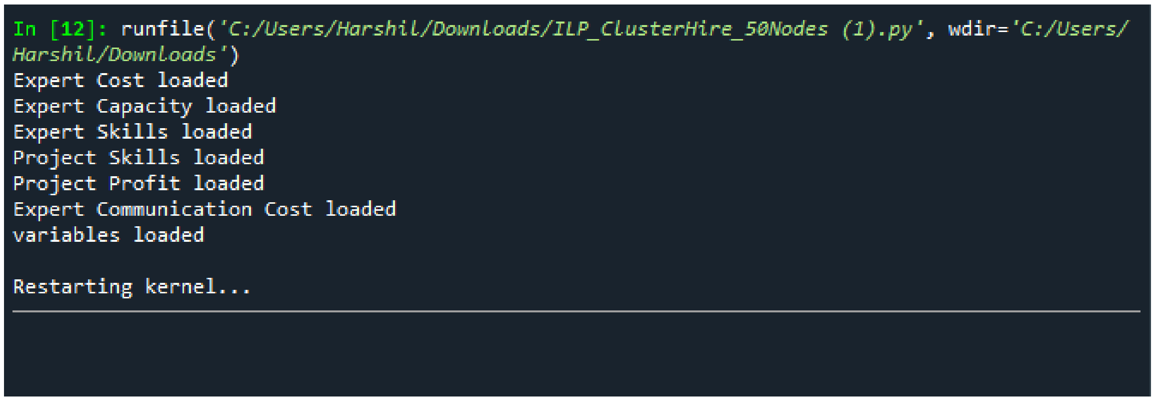


Figure 23: Total Profit vs Budget (Number of projects = 14)

4.4 Comparison with ILP approach

Along with the greedy algorithm, Integer Linear Programming (ILP) approach was also implemented in the past to solve the Cluster Hire problem in social networks. Sagarika et. al.[31] proposed ILP approach, which was able to generate near-optimal solutions compared to greedy heuristic solutions for the Cluster Hire problem in social networks. The drawbacks of the ILP approach are that it needs a huge amount of memory for variables and constraints creation and the runtime of the approach is also very high for large social networks. ILP traverses through all the possible best objective functions; therefore, runtime for ILP is generally very high compared to greedy solutions.

We run the ILP algorithm for a small dataset consisting of 50 experts, 26 skills, and 5 projects as used in research work[31]. We implemented the ILP approach on Spyder (Python 3.8) IDE using Anaconda Navigator. Our experiments were conducted on a computer with device specifications of Intel Core i7-6700 CPU @ 3.40 GHz, on a 64-bit Windows operating system with 8 GB RAM. Due to the ILP approach's computational complexity, it could not produce results on our system. The kernel crashes because it asks for an unreasonable amount of memory and terminates the program, as shown in Figure 24 below. The ILP program runs for almost 4 hours and still could not produce any results due to memory constraints.



```
In [12]: runfile('C:/Users/Harshil/Downloads/ILP_ClusterHire_50Nodes (1).py', wdir='C:/Users/
Harshil/Downloads')
Expert Cost loaded
Expert Capacity loaded
Expert Skills loaded
Project Skills loaded
Project Profit loaded
Expert Communication Cost loaded
variables loaded

Restarting kernel...
```

Figure 24: ILP Result (50 experts)

The minimum system requirement is of 32 GB RAM for the given input size, as stated in research work[31]. As we have limited RAM on our system, we could not run the ILP approach and perform the comparison with the MWSCAN algorithm even for a small dataset of 50 experts. But, on the same system MWSCAN algorithm was able to generate near-optimal solution to the Cluster Hire problem in social networks for a large dataset of 50K experts in a timely manner. As the ILP algorithm cannot generate results, we compared the results of the MWSCAN algorithm with a greedy algorithm.

4.5 Discussion

In this section, we will discuss the performance of our proposed algorithm to solve the cluster hire problem in social networks with an objective to select a group of experts within the given budget and working capacity, with the minimum communication cost between a group of experts. We proposed the MWSCAN algorithm to solve the Cluster Hire problem in social networks. We performed various experiments to evaluate the performance of our approach. We perform experiments on the five sets of projects having sizes 3, 5, 7, 10 and, 14. The results show that solving the cluster hire problem in social networks with the MWSCAN algorithm gives reliable and efficient results. It satisfies all the constraints of budget, communication cost, and skill requirements for the problem.

To evaluate the performance of our approach, we compared our results with the past greedy approach. We performed comparisons based on the number of projects completed vs. budget for two sets of projects having sizes 10 and 14. Also, we performed comparisons based on the total profit generated vs. budget for both the MWSCAN and greedy algorithm. Based on the results of the comparisons, MWSCAN produces more efficient results compared to the greedy algorithm. First of all, the MWSCAN algorithm completed a greater number of projects in the given budget as compared to the greedy algorithm. Also, the MWSCAN generated more profit for the given budget in comparison to the greedy algorithm. From the comparison results, it is clear that organizations will profit more if they use the MWSCAN algorithm to solve the Cluster Hire problem in social networks rather than using past greedy algorithms. Therefore, we can say that MWSCAN produces a more cost-efficient and profitable solution to the Cluster Hire problem in social networks compared to the greedy algorithm.

We also performed experiments to compare the runtime of both approaches for a different number of projects. We vary the number of skills required in each project between 3 to 7, and measure the runtime to get the average runtime for given number of projects. Figure 24 shows the average runtime comparison between our proposed approach MWSCAN and the previous greedy algorithm. As can be seen in Figure 2, the MWSCAN algorithm generates results in almost the same time as the greedy algorithm for three projects in a group. But as we increase the number of projects in our experiments, the average runtime of the MWSCAN algorithm increases significantly compared to the greedy algorithm. It can be considered as one limitation of the MWSCAN algorithm in comparison to the greedy algorithm. But, as we are running the experiments on a large dataset of 50K nodes, it is understandable that the runtime is a bit higher. Also, the MWSCAN algorithm searches the entire social network of experts to find the near-optimal solution to the Cluster Hire problem in social networks. In comparison, greedy algorithms do not generate a near-optimal solution and make locally optimal choices at each step.

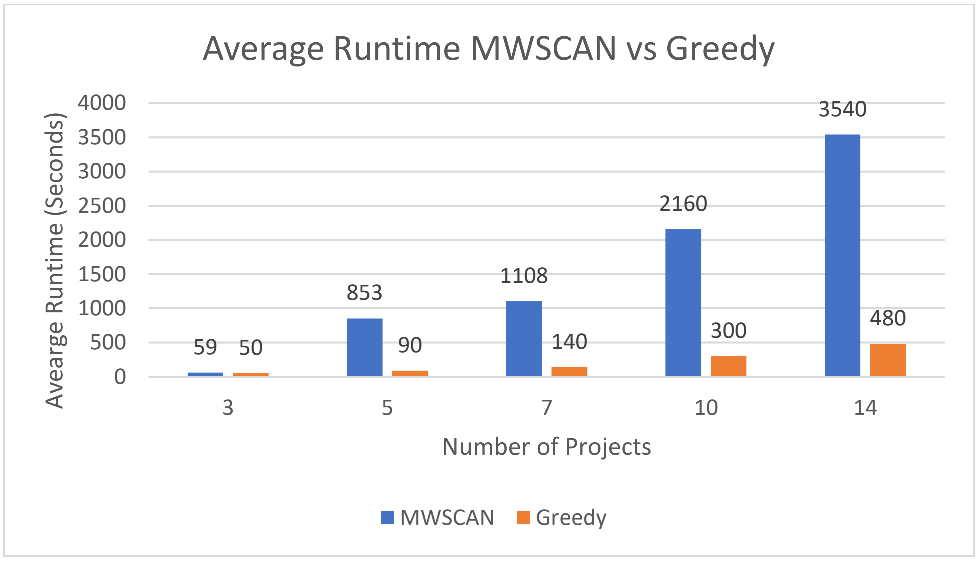


Figure 25: Average Runtime vs Number of projects

Our thesis aimed to study and analyze the impact of MWSCAN algorithm on solving the Cluster Hire problem in social networks. Cluster Hire problem in social networks is NP-hard in nature; it is hard to find the optimal/near-optimal solution due to all the constraints and objectives. The greedy solutions used in the past did not produce near-optimal results and failed to produce a feasible solution to the problem. From the comparison results, we can say that the MWSCAN algorithm could complete a greater number of projects and generate more profit for the given value of a budget compared to the greedy algorithm. Therefore, we can conclude that the MWSCAN algorithm produced an efficient, profitable, and near-optimal solution to the Cluster Hire problem in social networks in a reasonable amount of time.

CHAPTER 5
CONCLUSION AND FUTURE WORK

5.1 Conclusion

Several studies have been conducted in the past to address the Cluster Hire problem in social networks. In our study, we used the MWSCAN (Modified Weighted Structural Clustering Algorithm for Networks) to solve the Cluster Hire problem in social networks. The goal is to hire a group of experts within the given budget and working capacity while minimizing the communication cost among the hired group of experts. Hiring the group of experts within the given budget and working capacity of experts while minimizing the communication cost turns the problem into an NP-hard problem. We are the first to solve the problem of a cluster hire in social networks using a clustering algorithm. Our experiments on the DBLP dataset show that the MWSCAN algorithm can effectively select a group of experts with minimum communication cost for a given set of projects within the given constraints. The results produced by the MWSCAN are significantly effective compared to previous approaches.

Additionally, our study is derived from a more practical and realistic hiring scenario, which could be applied to almost all types of departments. It emphasizes saving time and money while hiring a collaborative group of experts, which helps with efficient working as a group and the finances of the company. We consider hiring a group of experts with multiple skills for different projects so that they can carry out several projects for the same organization, given that they have the required skills and the working capacity. This will help organizations to save on hiring cost for the set of projects.

For comparison with the MWSCAN algorithm, we use the greedy algorithm used in the past to solve the Cluster Hire problem in social networks. The results produced by the MWSCAN algorithm are better and more efficient than the greedy algorithm. The MWSCAN completed more projects and generated more profit for the organizations for the given value of a budget. The MWSCAN proved to be an effective solution to the cluster hire problem in social networks and generated near-optimal results. The runtime of our approach was a bit higher than the greedy algorithm due to multiple reasons. The greedy algorithm makes locally optimal choices at each step and does not look for a near-optimal solution; therefore, the runtime is low. In contrast, the

MWSCAN algorithm scans the entire network of experts to find the near-optimal solution for the Cluster Hire problem in social networks, leading to an increase in runtime. Also, the experiments were performed on the large dataset of 50K experts, which contributes to the increased runtime of algorithms.

5.2 Future Work

In the current setup, we consider communication cost, hiring cost, budget, and capacity as parameters while hiring a group of experts. We can extend the current work by adding more parameters to the cluster hire problem like experience level, work position level (i.e., manager, team lead), geographic location, etc. This can make the cluster hire problem more and more relatable for organizations in real life.

Currently, we are performing experiments on the DBLP dataset, which is static in nature, i.e., it is not changing with time. The application of MWSCAN on dynamic social networks can be seen as an extension of this research work. Dynamic social networks can provide more latest and relevant results.

BIBLIOGRAPHY

1. Scott, J. (1988). Social network analysis. *Sociology*, *22*(1), 109-127.
2. Barabási, A. L., & Albert, R. (1999). Emergence of scaling in random networks. *science*, *286*(5439), 509-512.
3. El-Sayed, A. M., Scarborough, P., Seemann, L., & Galea, S. (2012). Social network analysis and agent-based modeling in social epidemiology. *Epidemiologic Perspectives & Innovations*, *9*(1), 1-9.
4. Van der Hulst, R. C. (2009). Introduction to Social Network Analysis (SNA) as an investigative tool. *Trends in Organized Crime*, *12*(2), 101-121.
5. Kas, M., Carley, K. M., & Carley, L. R. (2012, April). Who was where, when? spatiotemporal analysis of researcher mobility in nuclear science. In *2012 IEEE 28th International Conference on Data Engineering Workshops* (pp. 347-350). IEEE.
6. Lappas, T., Liu, K., & Terzi, E. (2009, June). Finding a team of experts in social networks. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 467-476).
7. Wang, P., Xu, B., Wu, Y., & Zhou, X. (2015). Link prediction in social networks: the state-of-the-art. *Science China Information Sciences*, *58*(1), 1-38.
8. Bedi, P., & Sharma, C. (2016). Community detection in social networks. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *6*(3), 115-135.
9. Balabanović, M., & Shoham, Y. (1997). Fab: content-based, collaborative recommendation. *Communications of the ACM*, *40*(3), 66-72.
10. Xu, X., Yuruk, N., Feng, Z., & Schweiger, T. A. (2007, August). Scan: a structural clustering algorithm for networks. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 824-833).
11. Tran, T. N., Drab, K., & Daszykowski, M. (2013). Revised DBSCAN algorithm to cluster data with dense adjacent clusters. *Chemometrics and Intelligent Laboratory Systems*, *120*, 92-96.
12. Golshan, B., Lappas, T., & Terzi, E. (2014, August). Profit-maximizing cluster hires. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 1196-1205).

13. Chertov, A. (2012). *Extension of graph clustering algorithms based on SCAN method in order to target weighted graphs*. University of Windsor (Canada).
14. Chertov, A., Kobti, Z., & Goodwin, S. D. (2010, August). Weighted scan for modeling cooperative group role dynamics. In *Proceedings of the 2010 IEEE Conference on Computational Intelligence and Games* (pp. 17-22). IEEE.
15. Upadhyayula, S. (2016). *Dominance in multi-population cultural algorithms* (Doctoral dissertation, University of Windsor (Canada)).
16. Parikh, P. (2017). *Knowledge migration strategies for optimization of multi-population cultural algorithm* (Doctoral dissertation, University of Windsor (Canada)).
17. Holland, J. H. (1992). *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press.
18. Dorn, C., & Dustdar, S. (2010, October). Composing near-optimal expert teams: A trade-off between skills and connectivity. In *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"* (pp. 472-489). Springer, Berlin, Heidelberg.
19. Gómez, V., Kaltenbrunner, A., & López, V. (2008, April). Statistical analysis of the social network and discussion threads in slashdot. In *Proceedings of the 17th international conference on World Wide Web* (pp. 645-654).
20. Han, Y., Wan, Y., Chen, L., Xu, G., & Wu, J. (2017, May). Exploiting geographical location for team formation in social coding sites. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining* (pp. 499-510). Springer, Cham.
21. Selvarajah, K., Bhullar, A., Kobti, Z., & Kargar, M. (2018, May). WSCAN-TFP: weighted scan clustering algorithm for team formation problem in social network. In *The Thirty-First International Flairs Conference*.
22. Kargar, M., & An, A. (2011, October). 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* (pp. 985-994).
23. Patel, M., & Kargar, M. (2017). Cluster hire in a network of experts.
24. Gaston, M. E., Simmons, J., & Marie desJardins. (2004, July). Adapting Network Structure for Efficient Team Formation. In *AAAI Technical Report (2)* (pp. 1-8).
25. Backstrom, L., Huttenlocher, D., Kleinberg, J., & Lan, X. (2006, August). Group formation in large social networks: membership, growth, and evolution. In *Proceedings of the 12th*

ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 44-54).

26. Wi, H., Oh, S., Mun, J., & Jung, M. (2009). A team formation model based on knowledge and collaboration. *Expert Systems with Applications*, *36*(5), 9121-9134.
27. Baykasoglu, A., Dereli, T., & Das, S. (2007). Project team selection using fuzzy optimization approach. *Cybernetics and Systems: An International Journal*, *38*(2), 155-185.
28. Wang, X., Zhao, Z., & Ng, W. (2016). USTF: a unified system of team formation. *IEEE Transactions on Big Data*, *2*(1), 70-84.
29. Anagnostopoulos, A., Becchetti, L., Castillo, C., Gionis, A., & Leonardi, S. (2012, April). Online team formation in social networks. In *Proceedings of the 21st international conference on World Wide Web* (pp. 839-848).
30. Kargar, M., An, A., & Zihayat, M. (2012, September). Efficient bi-objective team formation in social networks. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases* (pp. 483-498). Springer, Berlin, Heidelberg.
31. Khandelwal, S. N. (2019). Building Teams of Experts using Integer Linear Programming.
32. Akiba, T., Iwata, Y., & Yoshida, Y. (2013, June). Fast exact shortest-path distance queries on large networks by pruned landmark labeling. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data* (pp. 349-360).
33. Cohen, E., Halperin, E., Kaplan, H., & Zwick, U. (2003). Reachability and distance queries via 2-hop labels. *SIAM Journal on Computing*, *32*(5), 1338-1355.
34. Gajewar, A., & Das Sarma, A. (2012, April). Multi-skill collaborative teams based on densest subgraphs. In *Proceedings of the 2012 SIAM international conference on data mining* (pp. 165-176). Society for Industrial and Applied Mathematics.
35. Ma, T., Wang, Y., Tang, M., Cao, J., Tian, Y., Al-Dhelaan, A., & Al-Rodhaan, M. (2016). LED: a fast overlapping communities detection algorithm based on structural clustering. *Neurocomputing*, *207*, 488-500.
36. Selvarajah, K., Zadeh, P. M., Kobti, Z., Palanichamy, Y., & Kargar, M. (2021). A unified framework for effective team formation in social networks. *Expert Systems with Applications*, *177*, 114886.

VITA AUCTORIS

NAME:	Harshil Patel
PLACE OF BIRTH:	Vadodara, Gujarat, India
YEAR OF BIRTH:	1996
EDUCATION:	Charotar University of Science and Technology, Gujarat, India 2013-2017 B. Tech.
	University of Windsor, Windsor, Ontario 2017-2021 M. Sc.